

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**FAILU AUGŠUPIELĀDES UN APSTRĀDES SISTĒMA IT
UZŅĒMUMAM**

KVALIFIKĀCIJAS DARBS

Autors: **Linards Losāns**

Studenta apliecības Nr.: 1117067

Darba vadītājs: Dr.sc.comp Māris Vītiņš

RĪGA 2019

ANOTĀCIJA

Kvalifikācijas darba mērķis ir izstrādāt failu augšupielādes un apstrādes sistēmu IT uzņēmumam.

Sistēma "Failu augšupielādes un apstrādes sistēma IT uzņēmumam" ir publiska tiešsaistes sistēma, kas sniedz iespēju ieinteresētajiem lietotājiem augšupielādēt savus radioloģijas izmeklējumus uzņēmuma datu bāzē. Sistēma ļauj elektroniski augšupielādēt radioloģijas izmeklējumus, kuri ir sastopami ZIP formātā vai ierakstīti CD diskā. Sistēma tika izstrādāta galvenokārt izmantojot Go programmēšanas valodu.

Atslēgvārdi: radioloģijas izmeklējumi, RIS, DICOM, PACS, Golang

ABSTRACT

“FILE UPLOAD AND PROCESSING SYSTEM FOR AN IT ENTERPRISE”

The goal of the project is to create a file upload and processing system for an IT enterprise.

System “File upload and processing system for an IT enterprise” is a public online system that allows users to upload their radiology studies directly to enterprise database. The system allows uploading of radiology studies in ZIP format, or studies that are written on a CD. System is mainly developed using Go programming language.

Key words: radiology studies, RIS, DICOM, PACS, Golang

SATURS

APZĪMĒJUMU SARAKSTS	5
IEVADS	6
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	7
1.1. Ievads	7
1.2. Vispārējais apraksts	8
1.3. Funkcionālās prasības	11
1.4. Nefunkcionālās prasības	50
2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	51
2.1. Ievads	51
2.2. Dekompozīcijas apraksts	52
2.3. Moduļu atkarības	54
2.4. Daļējs funkciju un lietotāja saskarnes projektējums	55
3. TESTĒŠANAS DOKUMENTĀCIJA	58
3.1. Testēšanas apraksts	58
3.2. Testpiemēri	58
3.3. Rezultātu kopsavilkums	60
4. PROJEKTA ORGANIZĀCIJA	61
5. KVALITĀTES NODROŠINĀŠANA	62
6. KONFIGURĀCIJAS PĀRVALDĪBA	63
7. DARBIETILPĪBAS NOVĒRTĒŠANA	64
SECINĀJUMI	65
IZMANTOTIE AVOTI	66
PIELIKUMI	67
1. Pielikums. Augšupielādes un apstrādes moduļa pirmkoda fragmenti	67
2. Pielikums. Importēšanas moduļa pirmkoda fragmenti	69

APZĪMĒJUMU SARAKSTS

Autentifikācija - lietotāja atpazīšanas process, kura gaitā noskaidro, vai lietotājs ir pilnvarots izmantot datus, programmas vai ierīces, kā arī noteiktus sistēmas darbības režīmus.

Autorizācija - process, kurā datorsistēma lietotājam nosaka noteiktas pilnvaras un resursus sistēmā. Tiek pielietots tikai tiem lietotājiem, kas jau ir veikuši autentifikāciju.

Apstrādātājs - no angļu valodas – “handler”

Cron - uzdevumu plānotājs, ko izmanto, lai periodiski veiktu uzdevumu izpildi noteiktā laikā.

DCMTK - dažādu rīku un bibliotēku kolekcija, kas ir saistīti ar DICOM standartu.

DICOM - starptautisks standarts, kā tiek pārvaldīti medicīnisko izmeklējumu attēlu dati.

DPD - datu plūsmu diagramma.

Git - versiju kontroles sistēma.

Komandu uzvedne - no angļu valodas - “command prompt”.

LU - Latvijas Universitāte.

PPA - programmatūras projektējuma apraksts.

PPS - programmatūras prasību specifikācija.

Storeescu - rīks DICOM failu importēšanai.

Staigātājfunkcija - Go programmēšanas valodas “filepath” bibliotēkas specializēta funkcija, kas tiek izsaukta pret katru failu norādītajā direktoriņā.

IEVADS

Kvalifikācijas darba mērķis ir izstrādāt failu augšupielādes un apstrādes sistēmu, ko ikdienā varēs izmantot lietotāji savām vajadzībām.

Tēmas aktualitāti nosaka tas, ka sistēma ir īpaši nepieciešama ārzemēs dzīvojošiem Latvijas valstspiederīgajiem. Veicot jebkādu radioloģijas izmeklējumu medicīnas iestādē, kura nav uzņēmuma partneris, pašlaik izmeklējuma rezultātus pacienti var saņemt 3 veidos:

- Izmeklējuma rezultātu tiek ierakstīti kompaktdiskā un atdoti pacientam.
- Izmeklējuma rezultātus var nosūtīt uz pacienta e-pasta adresi arhivētā veidā.
- Izmeklējuma rezultāti tiek uzglabāti medicīniskās iestādes sistēmā.

Gadījumos, kad izmeklējuma rezultāti ir ierakstīti kompaktdiskā vai nosūtīti uz e-pastu arhivētā veidā, izmeklējuma failu izkārtojums var stipri atšķirties, taču viena lieta tiem ir kopīga - izmeklējuma bildes ir DICOM formātā. Pacientam var rasties grūtības šos failus identificēt, tāpēc labākais risinājums būtu augšupielādēt visus izmeklējuma failus un ļaut sistēmai apstrādāt nepieciešamos.

Tā kā uzņēmuma partneri ir lielākā daļa Latvijā esošo medicīnas iestāžu, tad pēc izmeklējuma augšupielādēšanas pacienta ģimenes ārsts varēs pieslēgties uzņēmuma sistēmai un aplūkot veikto izmeklējumu.

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Šī programmatūras prasību specifikācija (PPS) ir izstrādāta un paredzēta sistēmas „Failu augšupielādes un apstrādes sistēma IT uzņēmumam” prasību aprakstīšanai. Atbilstoši šai specifikācijai tiks izstrādāta sistēma. Šis dokuments ir savstarpēja vienošanās starp pasūtītāju un izpildītāju.

Šī sistēma tika izstrādāta un tālākā dokumentācija tiks izstrādāta, vadoties pēc šīs specifikācijas. Šajā prasību specifikācijā tiek formulētas sistēmas prasības un raksturota funkcionalitāte.

1.1.2. Darbības sfēra

Programmatūra ir paredzēta lietošanai cilvēkiem, kuriem ir nepieciešamība attālināti augšupielādēt savus radioloģijas izmeklējumus.

1.1.3. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standartu LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis” [1] prasības un LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” [2].

1.1.4. Pārskats

Dokuments sastāv no 4 nodaļām:

1. Ievads. Nodaļā tiek aprakstīts dokumenta nolūks, produkta darbības sfēra, kā arī dokumenta struktūra un saistība ar citiem dokumentiem.
2. Vispārējais apraksts. Nodaļā tiek aprakstīts esošais stāvoklis, izklāstīta produkta perspektīva, funkcionalitāte, informācija par rīka lietotājiem, kā arī definēti rīka ierobežojumi, pieņēmumi un atkarības.
3. Funkcionālās prasības. Nodaļā tiek plašāk aprakstīta rīka funkcionalitāte, ievaddati, apstrāde, izvaddati un kļūdu paziņojumi.
4. Nefunkcionālās prasības. Nodaļa apraksta sistēmas drošības, veiktspējas, uzturamības un izmantojamības prasības.

1.2. Vispārējais apraksts

1.2.1. Esošā stāvokļa apraksts

Šobrīd uzņēmuma sistēmā lietotājam nav iespējas augšupielādēt radioloģijas izmeklējumus.

Radioloģijas izmeklējumu rezultāti no medicīnas iestādes var tikt saņemti četros veidos:

- var tikt izsniegts CD disks, kurā ir izmeklējuma rezultāti DICOM formātā;
- uz e-pastu var tikt nosūtīts ZIP fails, kurš satur izmeklējuma rezultātus DICOM formātā;
- izmeklējuma rezultāti tiek augšupielādēti medicīniskās iestādes mājaslapā;
- izmeklējuma rezultāti tiek automātiski augšupielādēti uzņēmuma datu bāzē, ja medicīniskā iestāde ir uzņēmuma partneris.

Pirmajos divos gadījumos izmeklējuma rezultāti ir saņemti, taču parastam lietotājam tie neko neizsaka. Gadījumā, ja izmeklējums tika veikts ārzemēs un tika izsniegts CD disks, tad, lai nosūtītu rezultātus savam ārstam Latvijā, var rasties grūtības, tāpēc ir nepieciešams izveidot sistēmu, kas šo problēmu atrisina.

1.2.2. Produkta perspektīva

Izstrādājamais produkts nevar darboties patstāvīgi, jo tiek pieņemts, ka jau ir eksistējošs PACS serveris, lai importētu apstrādātos failus. Izstrādājamā sistēma atradīsies un tiks darbināta no uzņēmuma serveriem. Sistēmas lietošanai būs nepieciešams interneta pieslēgums. Sistēmas galalietotājs var būt jebkurš Latvijas pilsonis, kuram no ārsta ir izsniegtas piekļuves tiesības, vai arī kuram ir konts Swedbank, SEB, Citadele, Nordea vai DNB bankā.

1.2.3. Sistēmas funkcijas

Failu augšupielādes un apstrādes sistēmai tiek nodrošinātas sekojošās funkcijas:

1. Augšupielādes un apstrādes modulis

- Inicializācijas funkcija
- Galvenā funkcija
- Augšupielādes servera funkcija
- Atlikušo failu pārbaudes iesaiņojuma funkcija
- Atlikušo failu pārbaudes funkcija
- Kļūdaino failu pārvietošanas funkcija
- Augšupielādes pabeigšanas funkcija
- Failu apstrādes funkcija

- Augšupielādēto failu dzēšanas funkcija
- Ekstraktēto failu apstrādes funkcija
- Faila tipa noskaidrošanas funkcija
- Izmeklējuma bildes identifikatora noskaidrošanas funkcija
- Izmeklējuma studijas identifikatora noskaidrošanas funkcija
- Dicom failu apstrādes funkcija
- Direktoriiju izveidošanas funkcija
- Indeksa funkcija
- Atslēgšanās funkcija
- Notifikāciju sūtīšanas funkcija
- Tiesību pārbaudes funkcija
- Sīkdatnes apstrādes funkcija
- Šodienas noskaidrošanas funkcija
- Kritiskas kļūdas funkcija
- Konfigurācijas lasīšanas funkcija
- ISO failu iesaiņojuma funkcija
- ISO failu ekstraktēšanas funkcija
- ZIP failu iesaiņojuma funkcija
- ZIP failu ekstraktēšanas funkcija

2. Importēšanas modulis

- Inicializācijas funkcija
- Galvenā funkcija
- Studijas apstrādes funkcija
- Studiju eksistēšanas pārbaudes funkcija
- Studiju identifikatora validācijas funkcija
- Studijas nosūtīšanas funkcija
- Kļūdaino studiju pārvietošanas funkcija
- Labo studiju pārvietošanas funkcija
- Direktoriiju izveidošanas funkcija
- Studiju notifikāciju sūtīšanas funkcija
- Dienas notifikāciju sūtīšanas funkcija
- Datora komandu uzvednes funkcija

1.2.4. Lietotāja raksturiezīmes

Sistēma ir paredzēta lietošanai. Lietotājiem nepieciešamas prasmes darbā ar datoru un ar uzstādīto Interneta pārlūkprogrammu. Lietotājam ir jābūt savam kontam Swedbank, SEB, Citadele, Nordea vai DNB bankā, vai arī jābūt no ārsta izsniegtiem pieslēgšanās datiem, lai varētu izmantot sistēmu.

1.2.5. Vispārējie ierobežojumi

1. Darbam ar sistēmu nepieciešams dators ar piekļuvi internetam.
2. Sistēmas darbībai jābūt atbalstītai uz četrām populārākajām pārlūkprogrammām: Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.
3. Sistēmai ir jāizmanto šifrēšana datu pārraidē, jo paredzēts, ka tā būs pieejama no publiskā tīkla.

1.2.6. Pieņēmumi un atkarības

Pieņēmumi faktoriem, kas spēj ietekmēt sistēmas programmatūras prasības:

- Sistēmas lietotāji programmatūras darbināšanai izmantos aktuālas pārlūkprogrammu versijas, jo, izmantojot vecās pārlūkprogrammu versijas, tas var ietekmēt programmatūras darbību un to vizuālo izskatu.
- Sistēmai tiks nodrošināts jau strādājošs PACS serveris.

1.3. Funkcionālās prasības

Failu augšupielādes un apstrādes sistēmas funkcijas pēc to nozīmes tiek iedalītas 2 moduļos:

- augšupielādes un apstrādes modulis, kurš sevī ietver failu augšupielādes un apstrādes funkcionalitāti;
- importēšanas modulis, kurš atbild par apstrādāto failu importēšanu uzņēmuma sistēmā.

1.3.1. Augšupielādes un apstrādes modulis

1.3.1.1.tabula Inicializācijas funkcija

Funkcijas nosaukums
init
Apraksts
“ init ” ir Golang programmēšanas valodai specializēta funkcija, kas tiek izpildīta pirms “ main ” funkcijas. Funkcija paredzēta, lai sagatavotu vidi turpmākai programmas darbībai.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none">• Tiek palaista “readconfig” funkcija, kas ieraksta konfigurācijas datus “conf” mainīgajā.• Tiek izveidots jauns Cron darbs, kas katru dienu pulksten 02:00 palaidīs “notify” funkciju.• Tiek palaista “makeDirs” funkcija, lai izveidotu nepieciešamās direktorijas.
Izvade
Nav
Kļūdas paziņojumi
<ol style="list-style-type: none">1. bad ConfigData2. bad NotifySchedule: <i>ERROR</i>3. failed to make directories

Funkcijas nosaukums
main
Apraksts
<p>“main” ir programmas galvenā funkcija. Tā tiek palaista uzreiz pēc “init” funkcijas. Funkcija paredzēta, lai palaistu visus programmas darbībai nepieciešamos procesus.</p>
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> • Tiek palaista “leftOverCheck” funkcija, kas apstrādā atlikušos augšupielādētos failus konfigurācijā norādītajā mapē. • Tiek palaista “tusdServer” funkcija, kas nodrošina failu augšupielādēšanas iespēju. • Tiek izveidots serveris, kuram tiek noteikti “/” un “/a/” segmenti, kas nodrošinās lapas rādīšanu un statisko failu padošanu pārlūkprogrammai.
Izvade
Nav.
Kļūdas paziņojumi
Nav.

Funkcijas nosaukums
tusdServer
Apraksts
Funkcija paredzēta, lai palaistu “tusd” bibliotēkas serveri, kas klausīsies norādīto portu un gaidīs POST pieprasījumus. Tusd serveris ir atbildīgs par failu augšupielādi.
Ievades parametri
composer
Ievades parametru tips
<i>*tusd.StoreComposer</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots jauns apstrādātājs, kas gaidīs POST pieprasījumus uz “/files/” segment. • Pieprasījuma saņemšanas brīdī tiek izsaukta “checkPermissions” funkcija, lai pārbaudītu, vai lietotājs, kurš sūtīja šo pieprasījumu, ir pierakstījies sistēmā. • Tiek palaists serveris uz konfigurācijā norādīto portu.
Izvade
Nav
Kļūdas paziņojumi
<ol style="list-style-type: none"> 1. Handler error: <i>ERROR</i> 2. Unable to listen: <i>ERROR</i> 3. CheckPermissions error at POST: <i>ERROR</i>

1.3.1.4.tabula Atlikušo failu pārbaudes iesaiņojuma funkcija

Funkcijas nosaukums
leftOverCheck
Apraksts
Funkcija izsauc funkciju “ walk ” uz konfigurācijā norādīto augšupielādes direktoriju.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none">• Katram failam norādītajā direktorijā tiek izsaukta funkcija “walk”.
Izvade
Nav
Kļūdas paziņojumi
1. <i>ERROR</i>

Funkcijas nosaukums
walk
Apraksts
Walk funkcija ir staigātājfunkcija, jo tā tiek izsaukta katram failam iepriekš norādītajā direktoriājā. Funkcija paredzēta, lai apstrādātu augšupielādes mapē pārpalikušos failus.
Ievades parametri
path, fi, err
Ievades parametru tips
<i>string, os.FileInfo, error</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek meklēts fails ar pagarinājumu “.info”. • No atrastā “.info” faila nolasa metadatus, no kuriem iegūst īsto binārā faila izmēru, kādam vajadzētu būt, kad izmeklējums ir pilnībā augšupielādēts. • Mēģina nolasīt failu ar pagarinājumu “.bin”, kura nosaukums ir tāds pats, kā iepriekš nolasītajam “.info” failam. • Salīdzina vai binārā faila izmērs sakrīt ar iepriekš iegūto lielumu. • Gadījumā, ja izmēri nesakrīt – tas nozīmē, ka fails vēl nav līdz galam augšupielādēts, tāpēc funkcija atgriež “nil” vērtību un meklē nākamo “.info” failu. • Gadījumā, ja izmēri sakrīt, tad tiek palaista “process” funkcija.
Izvade
<i>error</i>
Kļūdas paziņojumi
<ol style="list-style-type: none"> 1. failed to read .info file: <i>ERROR</i> (Path: <i>BINFILENAME</i>) 2. failed to parse .info file: <i>ERROR</i> (Path: <i>BINFILENAME</i>) 3. failed to get file stats: <i>ERROR</i> (Path: <i>BINFILENAME</i>) 4. failed to process file at leftOverCheck: <i>ERROR</i> (Path: <i>PATH</i>)

1.3.1.6.tabula Kļūdaino failu pārvietošanas funkcija

Funkcijas nosaukums
moveToBadDir
Apraksts
Funkcija paredzēta, lai pārvietotu augšupielādētos failus uz kļūdu direktoriju.
Ievades parametri
badDirName, filePath
Ievades parametru tips
<i>string, string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidota jauna direktorija ar nosaukumu no mainīgā “badDirName” vērtības. • Tiek pārvietots “.bin” fails uz jaunizveidoto direktoriju. • Tiek pārvietots “.info” fails uz jaunizveidoto direktoriju.
Izvade
<i>error</i>
Kļūdas paziņojumi
<ol style="list-style-type: none"> 5. failed to create baddir: <i>ERROR</i> 6. failed to move to baddir: <i>ERROR</i>

1.3.1.7.tabula Augšupielādes pabeigšanas funkcija

Funkcijas nosaukums
FinishUpload
Apraksts
Funkcija tiek izsaukta brīdī, kad tiek pabeigta faila augšupielāde.
Ievades parametri
id
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek palaista “process” funkcija, lai apstrādātu augšupielādētos failus. Ja “process” funkcija atgriež kļūdu, tad šīs dienas kļūdu skaits tiek palielināts par 1 un tiek izsaukta “moveToBadDir” funkcija, lai pārvietotu failus uz kļūdu direktoriju. • Tiek palaista “removeUploaded” funkcija.
Izvade
<i>error</i>
Kļūdas paziņojumi
1. failed to process file at FinishUpload: <i>ERROR</i> (Path: <i>PATH</i>)

Funkcijas nosaukums
process
Apraksts
Funkcija tiek izsaukta augšupielādētajiem failiem. Funkcija paredzēta, lai sagatavotu augšupielādētos failus importēšanai uzņēmuma sistēmā, izsaucot dažādas citas funkcijas, kas apstrādā un atlasa nepieciešamos failus.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izsaukta “findType” funkcija, lai noskaidrotu “.bin” faila tipu. • Ja faila tips ir “zip”, tad tiek izsaukta “unZIP” funkcija, lai failu ekstraktētu. • Ja faila tips ir “iso”, tad tiek izsaukta “unISO” funkcija, lai failu ekstraktētu. • Tiek izsaukta “processExtracted” staigātājfunkcija. • Tiek izdzēsti pēc apstrādes atlikušie faili.
Izvade
<i>error</i>
Kļūdas paziņojumi
<ul style="list-style-type: none"> • unZIP remove error: <i>ERROR</i> • unISO remove error: <i>ERROR</i> • processExtracted remove error: <i>ERROR</i>

1.3.1.9.tabula Augšupielādēto failu dzēšanas funkcija

Funkcijas nosaukums
removeUploaded
Apraksts
Funkcija paredzēta, lai izdzēstu augšupielādētos failus pēc to apstrādāšanas.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izdzēsts “.info” fails. • Tiek izdzēsts “.bin” fails.
Izvade
<i>error</i>
Kļūdas paziņojumi
1. Upload remove error: <i>ERROR</i>

1.3.1.10.tabula Ekstraktēto failu apstrādes funkcija

Funkcijas nosaukums
processExtracted
Apraksts
Funkcija paredzēta, lai no visiem ekstraktētajiem failiem atlasītu vajadzīgos, kurus pēc tam padotu vēl tālākai apstrādei.
Ievades parametri
path, fi, err
Ievades parametru tips
<i>string, os.FileInfo, error</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izsaukta “findType” funkcija, lai noskaidrotu faila tipu. • Ja faila tips nav “dcm”, tad funkcija pāriet pie nākamā faila. • Ja faila tips ir “dcm”, tad tiek izsaukta funkcija “processDICOM”.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

1.3.1.11.tabula **Faila tipa noskaidrošanas funkcija**

Funkcijas nosaukums
findType
Apraksts
Funkcija paredzēta, lai noskaidrotu faila tipu, pārbaudot faila “magiskos skaitļus” jeb faila parakstu.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts fails. • Mēģina nolasīt pirmos 37000 baitus. • Salīdzina noteiktu baitu vērtības. • Atgriež rezultātu.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

1.3.1.12.tabula Izmeklējuma bildes identifikatora noskaidrošanas funkcija

Funkcijas nosaukums
getInstanceUID
Apraksts
Funkcija paredzēta, lai noskaidrotu izmeklējuma bildes unikālo identifikatoru, kas ir atrodams DICOM faila metadatos.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts DICOM fails. • Tiek atrasta “SOPInstanceUID” vērtība. • Atgriež iegūto vērtību.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
1. failed to read/parse dicom file: <i>ERROR</i> (Path: <i>PATH</i>)

1.3.1.13.tabula Izmeklējuma studijas identifikatora noskaidrošanas funkcija

Funkcijas nosaukums
getStudyUID
Apraksts
Funkcija paredzēta, lai noskaidrotu izmeklējuma studijas unikālo identifikatoru, kas ir atrodams DICOM faila metadatos.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts DICOM fails. • Tiek atrasta “SOPStudyUID” vērtība. • Atgriež iegūto vērtību.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

1.3.1.14.tabula **Dicom failu apstrādes funkcija**

Funkcijas nosaukums
processDICOM
Apraksts
Funkcija tiek izsaukta visiem atrastajiem DICOM tipa failiem. Funkcija paredzēta, lai no DICOM faila dabūtu studijas identifikatoru un bildes identifikatoru un pārvietotu failus tiem atbilstošās studijas direktorijās.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izsaukta “getStudyUID” funkcija, kas atgriež studijas identifikatoru. • Tiek izsaukta “getInstanceUID” funkcija, kas atgriež bildes identifikatoru. • Tiek izveidota jauna direktorija studijas identifikatora vārdā, ja tāda vēl neeksistē. • DICOM tipa faili tiek pārvietoti uz studijas direktoriju. • Tiek palielināts šīs dienas apstrādāto DICOM failu skaits par 1.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

1.3.1.15.tabula **Direktoriju izveidošanas funkcija**

Funkcijas nosaukums
makeDirs
Apraksts
Funkcija paredzēta, lai pirms programmas darbības sākšanas tiktu izveidotas visas programmas darbībai nepieciešamās direktorijas.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> • Izveido augšupielādēšanas direktoriju. • Izveido ekstraktēšanas direktoriju. • Izveido DICOM failu uzturēšanas direktoriju. • Izveido kļūdu direktoriju, kur tiek pārlikti faili kļūdas gadījumā.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
handleIndex
Apraksts
Funkcija paredzēta, lai sniegtu lietotājam mājaslapas saturu.
Ievades parametri
w, r
Ievades parametru tips
<i>http.ResponseWriter, *http.Request</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izsaukta “checkPermissions” funkcija, lai noskaidrotu, vai lietotājs ir pieslēdzies sistēmai. • Ja lietotājs nav pieslēdzies sistēmai, lietotājs tiek novirzīts uz pieslēgšanās lapu. • Ja lietotājs ir pieslēdzies sistēmai, tiek parādīts lapas saturs.
Izvade
Nav
Kļūdas paziņojumi
1. CheckPermissions error at GET: <i>ERROR</i>

Funkcijas nosaukums
logoutHandler
Apraksts
Funkcija paredzēta, lai atslēgtu lietotāju no piekļuves sistēmas mājaslapai.
Ievades parametri
w, r
Ievades parametru tips
<i>http.ResponseWriter, *http.Request</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidota sīkdatne “dms” ar tukšu vērtību un dzīves ilgumu 0. • Lietotājs tiek novirzīts uz pieslēgšanās lapu.
Izvade
Nav
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
notify
Apraksts
Funkcija paredzēta, lai nosūtītu statistikas informāciju par iepriekšējo dienu.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots string tipa mainīgais, kas sevī ietvers informāciju par apstrādāto ZIP un ISO failu skaitu, kā arī kopējo pabeigto augšupielāžu skaitu. • Ja dienas laikā ir notikusi kaut viena kļūda failu apstrādes laikā, tad mainīgajam klāt tiek konkatēnēts neizdevušos apstrādāto failu skaits. • Ziņa tiek nosūtīta.
Izvade
Nav
Kļūdas paziņojumi
1. notify error: <i>ERROR</i>

Funkcijas nosaukums
checkPermissions
Apraksts
Funkcija paredzēta, lai noskaidrotu, vai lietotājs ir pieslēdzies sistēmā.
Ievades parametri
r
Ievades parametru tips
<i>*http.Request</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izsaukta “getCookie” funkcija, lai noskaidrotu, vai lietotājs ir pieslēdzies sistēmā. • Tiek pārbaudīts, vai lietotāja pieslēguma sesija vēl nav beigusies. • Atgriež rezultātu.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
getCookie
Apraksts
Funkcija paredzēta, lai no pieprasījuma dabūtu “dms” sīkdatni.
Ievades parametri
r
Ievades parametru tips
<i>*http.Request</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek pieprasīta “dms” sīkdatne. • Sīkdatne tiek sadalīta divās daļās pēc “=” dalītājzīmes. • Atgriež sīkdatnes pirmo daļu, kas satur sesijas ID.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
today
Apraksts
Funkcija paredzēta, lai noskaidrotu šodienas datumu.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none">• Tiek atgriezts rezultāts.
Izvade
<i>time.Time</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
fatal
Apraksts
Funkcija paredzēta, lai izdrukātu kļūdas paziņojumu un apturētu programmas darbību.
Ievades parametri
d
Ievades parametru tips
<i>...interface{}</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izdrukāts kļūdas paziņojums. • Tiek apturēta programmas darbība.
Izvade
Nav
Kļūdas paziņojumi
1. <i>ERROR</i>

Funkcijas nosaukums
readConfig
Apraksts
Funkcija paredzēta, lai nolasītu konfigurācijā norādītos datus.
Ievades parametri
assets, conf
Ievades parametru tips
<i>func(string)([]byte, error), ...interface{}</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts konfigurācijas fails. • Konfigurācijas dati tiek ierakstīti mainīgajā, kura struktūra sakrīt ar konfigurācijas faila struktūru.
Izvade
Nav
Kļūdas paziņojumi
<ol style="list-style-type: none"> 1. Asset error: <i>ERROR</i> 2. Config unmarshal error: <i>ERROR</i>

Funkcijas nosaukums
unISO
Apraksts
Funkcija paredzēta, lai sagatavotu ISO failu tālākai ekstraktēšanai.
Ievades parametri
src, destPath
Ievades parametru tips
<i>string, string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts ISO fails. • Tiek izveidots jauns ISO failu lasītājs. • Tiek izsaukta “extractAndWriteFileISO” funkcija.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
extractAndWriteFileISO
Apraksts
Funkcija paredzēta, lai ekstraktētu failus, kas ir arhivēti ISO formāta failā.
Ievades parametri
isoReader, destPath
Ievades parametru tips
<i>*iso9660.Reader, string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērta fails. • Ja fails ir direktorijs, tad tiek izveidota jauna direktorijs ar tādu pašu nosaukumu. • Ja fails nav direktorijs, tad tas tiek pārkopēts uz ceļu, kas norādīts destPath mainīgajā.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
unZIP
Apraksts
Funkcija paredzēta, lai sagatavotu ZIP failu tālākai ekstraktēšanai.
Ievades parametri
src, destPath
Ievades parametru tips
<i>string, string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots jauns ZIP failu lasītājs. • Tiek izsaukta “extractAndWriteFileZIP” funkcija.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
extractAndWriteFileZIP
Apraksts
Funkcija paredzēta, lai ekstraktētu failus, kas ir arhivēti ZIP formāta failā.
Ievades parametri
isoReader, destPath
Ievades parametru tips
*zip.File, string
Apstrāde
<ul style="list-style-type: none"> • Tiek atvērts fails. • Ja fails ir direktorija, tad tiek izveidota jauna direktorija ar tādu pašu nosaukumu. • Ja fails nav direktorija, tad tas tiek pārkopēts uz ceļu, kas norādīts destPath mainīgajā.
Izvade
error
Kļūdas paziņojumi
Nav

1.3.2. Importēšanas modulis

1.3.2.1.tabula **Inicializācijas funkcija**

Funkcijas nosaukums
init
Apraksts
Funkcija paredzēta, lai sagatavotu vidi turpmākai programmas darbībai.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none">• Tiek palaista “readconfig” funkcija, kas ieraksta konfigurācijas datus “conf” mainīgajā.• Tiek izveidots jauns Cron darbs, kas katru dienu pulksten 02:00 palaidīs “notify” funkciju.• Tiek palaista “makeDirs” funkcija, lai izveidotu nepieciešamās direktorijas.
Izvade
Nav
Kļūdas paziņojumi
<ol style="list-style-type: none">1. bad ConfigData2. bad RunEvery3. bad NotifySchedule: <i>ERROR</i>4. failed to make directories: <i>ERROR</i>

Funkcijas nosaukums
main
Apraksts
Funkcija paredzēta, lai pēc noteikta laika nolasītu DICOM failu direktorijas saturu un failu sastapšanas gadījumā izsauktu apstrādes funkcijas.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> ● Tiek izveidots bezgalīgs cikls. ● Tiek nolasīts konfigurācijā norādītās direktorijas saturs. ● Ja direktorijā ir sastopami faili, tad tiek izsaukta “processStudy” funkcija. ● Ja “processStudy” atgriež kļūdu, tad visa studija tiek pārvietota uz kļūdu direktoriju, kā arī dienas kļūdu skaits tiek palielināts par 1. ● Ja “processStudy” neatgriež kļūdu, tad visa studija tiek pārvietota uz apstrādāto studiju direktoriju. ● Cikla darbība tiek apturēta uz konfigurācijā norādītu laika daudzumu.
Izvade
Nav.
Kļūdas paziņojumi
<ol style="list-style-type: none"> 1. failed to read DICOM directory: <i>ERROR</i> 2. failed to process study: <i>ERROR (Study: STUDYNAME)</i> 3. failed to move to badDir: <i>ERROR (Study: STUDYNAME)</i> 4. failed to move to completeDir: <i>ERROR (Study: STUDYNAME)</i>

Funkcijas nosaukums
processStudy
Apraksts
Funkcija paredzēta, lai apstrādātu sagatavotās studijas.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek iegūts norādītās studijas saturs. • Tiek izveidots cikls, kurš iet pāri visiem failiem studijā. • Tiek nolasīts fails. • Pirmajā iterācijā tiek uzstādītas pamata “uid”, “institution” un “patName” mainīgo vērtības no nolasītā faila. • Pārējās iterācijās tiek salīdzināts, vai šīs vērtības sakrīt. • Tiek izsaukta “findStudy” funkcija, kurai padod “uid” mainīgo kā parametru. • Pārbauda, vai sistēmā jau eksistē tāda studija. • Ja neeksistē, tad tiek izsaukta “send” funkcija, kas sūtīs uzreiz visu studiju. • Ja eksistē, tad sūta katru studijas failu atsevišķi, vispirms pārbaudot, vai tas jau ir sastopams sistēmā. • Tiek izsaukta “studyNotify” funkcija.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

1.3.2.4.tabula Studiju eksistēšanas pārbaudes funkcija

Funkcijas nosaukums
findStudy
Apraksts
Funkcija paredzēta, lai pārbaudītu studijas eksistēšanu sistēmā.
Ievades parametri
studyUID
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots DICOM vaicājums, kur tiek norādīts studijas identifikators. • Nosūta vaicājumu un atgriež atbildi.
Izvade
<i>[]dicom.Tags, error</i>
Kļūdas paziņojumi
Nav

1.3.2.5.tabula **Studiju identifikatora validācijas funkcija**

Funkcijas nosaukums
isValidUID
Apraksts
Funkcija paredzēta, lai pārbaudītu studiju identifikatora autentiskumu.
Ievades parametri
studyUID
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> ● Tiek pārbaudīts, vai <ul style="list-style-type: none"> ○ identifikatora garums ir vismaz 20 zīmes, ○ identifikators satur vismaz 3 punktus, ○ identifikatora garums nepārsniedz 64 zīmes. ● Ja kaut viens no šiem kritērijiem neizpildās, tad tiek atgriezta vērtība “false”. Ja visi izpildās, tad atgriež vērtību “true”.
Izvade
<i>bool</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
send
Apraksts
Funkcija paredzēta, lai nosūtītu studiju vai atsevišķus studijas failus sistēmai.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • No konfigurācijas tiek nolasīti servera dati, uz kuru faili būs jāsūta. • Ja norādītais fails nav direktorijs, tad tiek izsaukta "localExec" funkcija. • Ja norādītais fails ir direktorijs, tad tiek izsaukta "localExec" funkcija ar papildus parametru "+sd", lai sūtītu visu studijas saturu.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

1.3.2.7.tabula Kļūdaino studiju pārvietošanas funkcija

Funkcijas nosaukums
moveToBadDir
Apraksts
Funkcija paredzēta, lai pārvietotu failus uz kļūdu direktoriju.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Studija tiek pārvietota uz konfigurācijā norādīto kļūdu direktoriju.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
moveToCompleteDir
Apraksts
Funkcija paredzēta, lai pārvietotu failus uz uzglabāšanas direktoriju.
Ievades parametri
path
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none">• Studija tiek pārvietota uz konfigurācijā norādīto uzglabāšanas direktoriju.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

1.3.2.9.tabula **Direktoriju izveidošanas funkcija**

Funkcijas nosaukums
makeDirs
Apraksts
Funkcija paredzēta, lai pirms programmas palaišanas tiktu izveidotas visas programmas darbībai nepieciešamās direktorijas.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> • Izveido uzglabāšanas direktoriju. • Izveido kļūdu direktoriju.
Izvade
<i>error</i>
Kļūdas paziņojumi
Nav

1.3.2.10.tabula Studiju notifikāciju sūtīšanas funkcija

Funkcijas nosaukums
studyNotify
Apraksts
Funkcija paredzēta, lai par katru importēto studiju tiktu nosūtīts paziņojums.
Ievades parametri
studyUID
Ievades parametru tips
<i>string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots “string” tipa mainīgais, kurš satur informāciju par datumu un studijas identifikatoru, kuru importēja. • Tiek nosūtīta notifikācija.
Izvade
Nav
Kļūdas paziņojumi
Nav

1.3.2.11.tabula Dienas notifikāciju sūtīšanas funkcija

Funkcijas nosaukums
notify
Apraksts
Funkcija paredzēta, lai katru dienu konfigurācijā norādītā laikā nosūtītu notifikāciju par dienas statistiku.
Ievades parametri
Nav
Apstrāde
<ul style="list-style-type: none"> • Tiek izveidots “string” tipa mainīgais, kurš satur informāciju par datumu, importēto studiju skaitu un kļūdu skaitu pagājušajā dienā. • Tiek nosūtīta notifikācija.
Izvade
Nav
Kļūdas paziņojumi
Nav

Funkcijas nosaukums
localExec
Apraksts
Funkcija paredzēta, lai programmai būtu iespēja izsaukt dažādas datora komandas programmas no komandu uzvednes.
Ievades parametri
cmd, args ...
Ievades parametru tips
<i>string, string</i>
Apstrāde
<ul style="list-style-type: none"> • Tiek palaista datora komanda “cmd” ar parametriem “args”. • Atgriež komandas atbildi.
Izvade
<i>string, error</i>
Kļūdas paziņojumi
Nav

1.4. Nefunkcionālās prasības

1.4.1. Veiktspēja

Tā kā augšupielādēto studiju izmēri var svārstīties no dažiem megabaitiem līdz pat vairākiem gigabaitiem, ir grūti noteikt laiku, kādā vienai studijai ir jābūt apstrādātai. Tiek prognozēts, ka sistēma lielāko laika daļu nebūs noslogota, tāpēc vienas studijas apstrādes laiks nav tik būtisks, bet tomēr pēc neformāliem pieņēmumiem tiek uzskatīts, ka vidēji vienas studijas apstrādes laikam nevajadzētu pārsniegt 5 sekundes.

1.4.2. Drošība

Piekļuve sistēmai tiek nodrošināta tikai autentificētiem lietotājiem. Mājaslapas piekļuvei sistēma izmantos HTTPS.

1.4.3. Uzturamība

Sistēmai ir jābūt izstrādātai tā, ka nepieciešamības gadījumā ir jābūt iespējai to papildināt ar jaunu funkcionalitāti, vai arī izmainīt esošo, minimāli izdarot izmaiņas pirmkodā. Šī iemesla dēļ sistēma tiek iedalīta divos moduļos, kuri atbildīs noteiktu darbību izpildīšanai. Funkcijām ir jābūt rakstītām tā, lai tās varētu viegli izmantot arī citos projektos, veicot minimālas izmaiņas pirmkodā. Sistēmas dokumentācijai vajag atbilst jaunākajai rīka versijai. Ja sistēmā tiek veiktas kādas izmaiņas, tad jāatjaunina arī dokumentācija.

1.4.4. Izmantojamība

Sistēmai ir jābūt intuiīvai saskarnei, kā rezultātā lietotājiem nevajadzētu rasties grūtībām to izmantot. Ir jābūt iespējai palaist sistēmu citā datorā, kur sistēma netika veidota vai testēta. Sistēmai var būt atkarības, taču to skaitam ir jābūt pēc iespējas mazākam.

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

2.1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt sistēmas projektējumu atbilstoši izvirzītajām prasībām. Šis dokuments ir paredzēts apraksītās sistēmas izstrādātājiem.

2.1.2. Darbības sfēra

Sistēma ir uzbūvēta ar Go, HTML, CSS, JavaScript palīdzību. Sistēmas lietotājs var būt jebkurš Latvijas iedzīvotājs, kuram no ārsta ir izsniegtas piekļuves tiesības, vai arī kāds, kuram ir konts Swedbank, SEB, Citadele, Nordea vai DNB bankai.

Programmatūra ir paredzēta lietošanai cilvēkiem, kuriem ir nepieciešamība attālināti augšupielādēt savus radioloģijas izmeklējumus.

2.1.3. Saistība ar citiem dokumentiem

Dokuments tika izstrādāts un noformēts, ievērojot standarta LVS 72:1996 "Ieteicamā prakse programmatūras projektējumam aprakstīšanai" [2] prasības un rekomendācijas.

2.1.4. Pārskats

Dokuments sastāv no 4 nodaļām:

1. Ievads. Nodaļā tiek aprakstīts dokumenta nolūks, produkta darbības sfēra, kā arī dokumenta struktūra un saistība ar citiem dokumentiem.
2. Dekompozīcijas apraksts. Nodaļā tiek aprakstīti rīka moduļi, to nolūks un funkcijas.
3. Moduļu atkarības. Nodaļā tiek attēlotas moduļu savstarpējās atkarības, izmantojot datu plūsmu diagrammas.
4. Daļējs funkciju un lietotāja saskarnes projektējums. Nodaļa satur diagrammas, kuras ļauj labāk izprast sistēmas darbību.

2.2. Dekompozīcijas apraksts

2.2.1. Moduļu dekompozīcija

Šajā nodaļā ir norādīts moduļu sadalījums sistēmā, to nolūks sistēmas darbībā, kā arī PPS realizācija. Moduļu savstarpējās atkarības ir aprakstītas nākamajā nodaļā.

2.2.1.1.tabula Augšupielādes un apstrādes modulis

Moduļa nolūks	Šis modulis atbild par augšupielādes servera nodrošināšanu, tiesību pārbaudi, notifikāciju sūtīšanu un augšupielādēto failu apstrādi.
Moduļa funkcijas	<ol style="list-style-type: none">1. Inicializācijas funkcija (init)2. Galvenā funkcija (main)3. Augšupielādes servera funkcija (tusdServer)4. Atlikušo failu pārbaudes iesaiņojuma funkcija (leftOverCheck)5. Atlikušo failu pārbaudes funkcija (walk)6. Kļūdaino failu pārvietošanas funkcija (moveToBadDir)7. Augšupielādes pabeigšanas funkcija (FinishUpload)8. Failu apstrādes funkcija (process)9. Augšupielādēto failu dzēšanas funkcija (removeUploaded)10. Ekstraktēto failu apstrādes funkcija (processExtracted)11. Faila tipa noskaidrošanas funkcija (findType)12. Izmeklējuma bildes identifikatora noskaidrošanas funkcija (getInstanceUID)13. Izmeklējuma studijas identifikatora noskaidrošanas funkcija (getStudyUID)14. DICOM failu apstrādes funkcija (processDICOM)15. Direktoriiju izveidošanas funkcija (makeDirs)16. Indeksa funkcija (handleIndex)17. Atslēgšanās funkcija (logoutHandler)18. Notifikāciju sūtīšanas funkcija (notify)19. Tiesību pārbaudes funkcija (checkPermissions)20. Sīkdatnes apstrādes funkcija (getCookie)21. Šodienas noskaidrošanas funkcija (today)

	<p>22. Kritiskas kļūdas funkcija (fatal)</p> <p>23. Konfigurācijas lasīšanas funkcija (readConfig)</p> <p>24. ISO failu iesaiņojuma funkcija (unISO)</p> <p>25. ISO failu ekstraktēšanas funkcija (extractAndWriteISO)</p> <p>26. ZIP failu iesaiņojuma funkcija (unZIP)</p> <p>27. ZIP failu ekstraktēšanas funkcija (extractAndWriteISO)</p>
--	--

2.2.1.2.tabula **Importēšanas modulis**

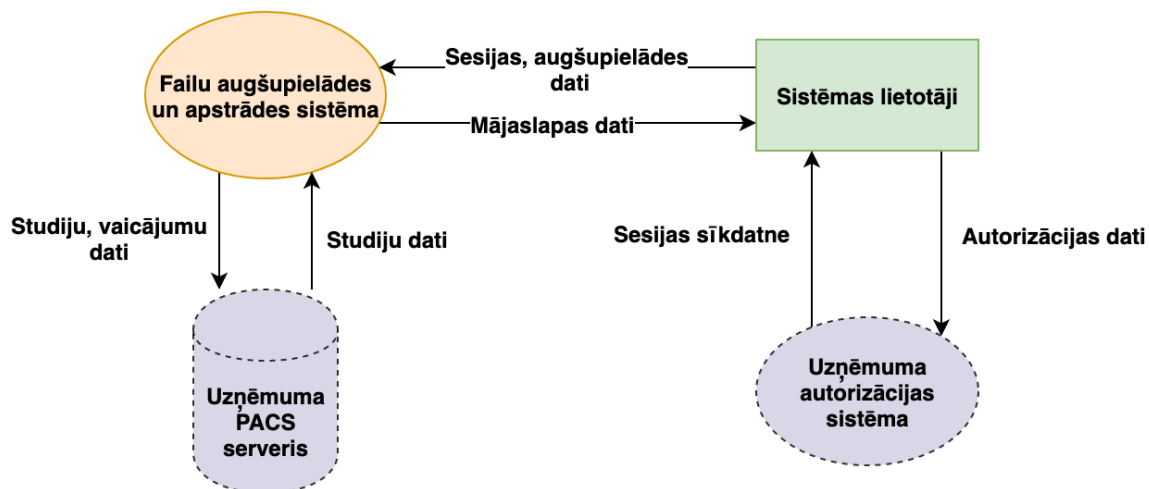
Moduļa nolūks	Šis modulis ir paredzēts, lai verificētu no iepriekšējās apstrādes atrastos failus, noskaidrotu, vai tie jau eksistē uzņēmuma sistēmā, un nosūtītu tos, kuri neeksistē.
Moduļa funkcijas	<ol style="list-style-type: none"> 1. Inicializācijas funkcija (init) 2. Galvenā funkcija (main) 3. Studijas apstrādes funkcija (processStudy) 4. Studiju eksistēšanas pārbaudes funkcija (findStudy) 5. Studiju identifikatora validācijas funkcija (isValidUID) 6. Studijas nosūtīšanas funkcija (send) 7. Kļūdaino studiju pārvietošanas funkcija (moveToBadDir) 8. Labo studiju pārvietošanas funkcija (moveToCompleteDir) 9. Direktoriju izveidošana funkcija (makeDirs) 10. Studiju notifikāciju sūtīšanas funkcija (studyNotify) 11. Dienas notifikāciju sūtīšanas funkcija (notify) 12. Datora komandu uzvednes funkcija (localExec)

2.2.2. Datu dekompozīcija

Veidotajā sistēmā nav paredzēts veidot jaunu datu bāzi, bet gan izmantot uzņēmumam jau esošo PACS serveri.

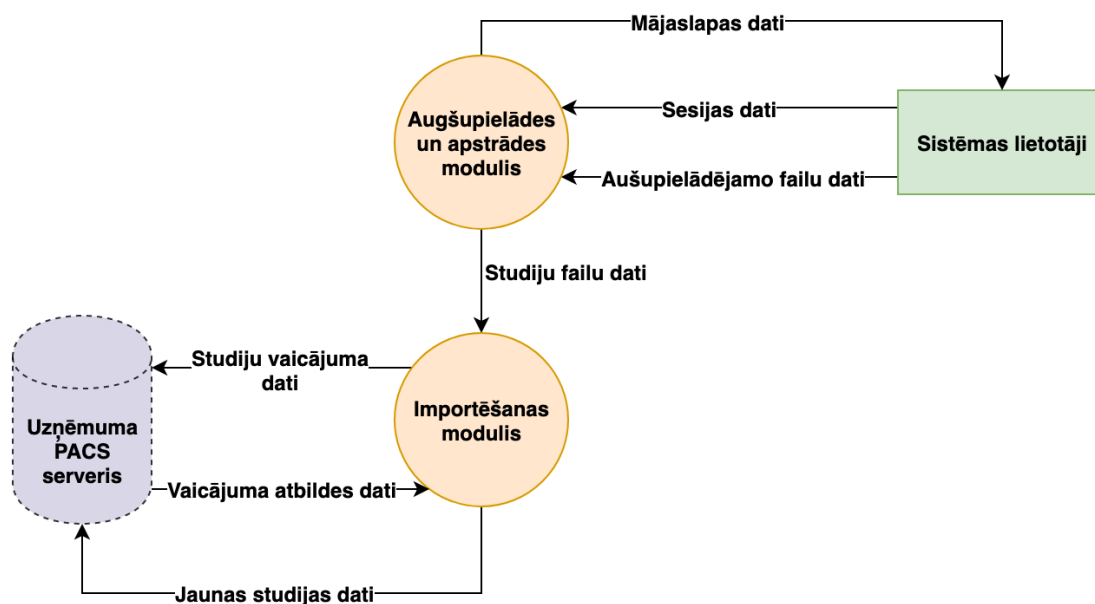
2.3. Moduļu atkarības

2.3.1. 0. līmeņa DPD



2.1.att. 0. līmeņa DPD

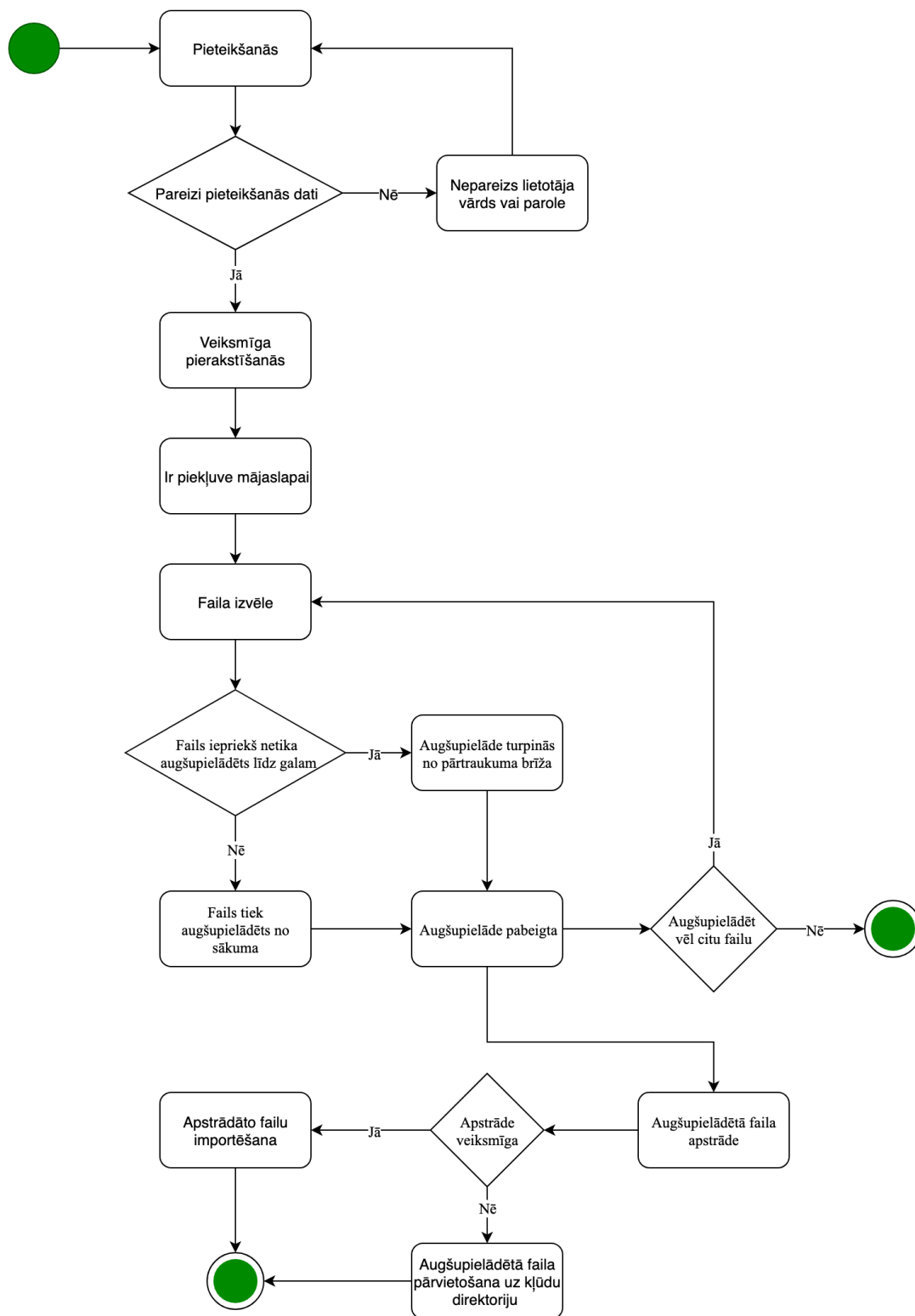
2.3.2. 1. līmeņa DPD



2.2.att. 1. līmeņa DPD

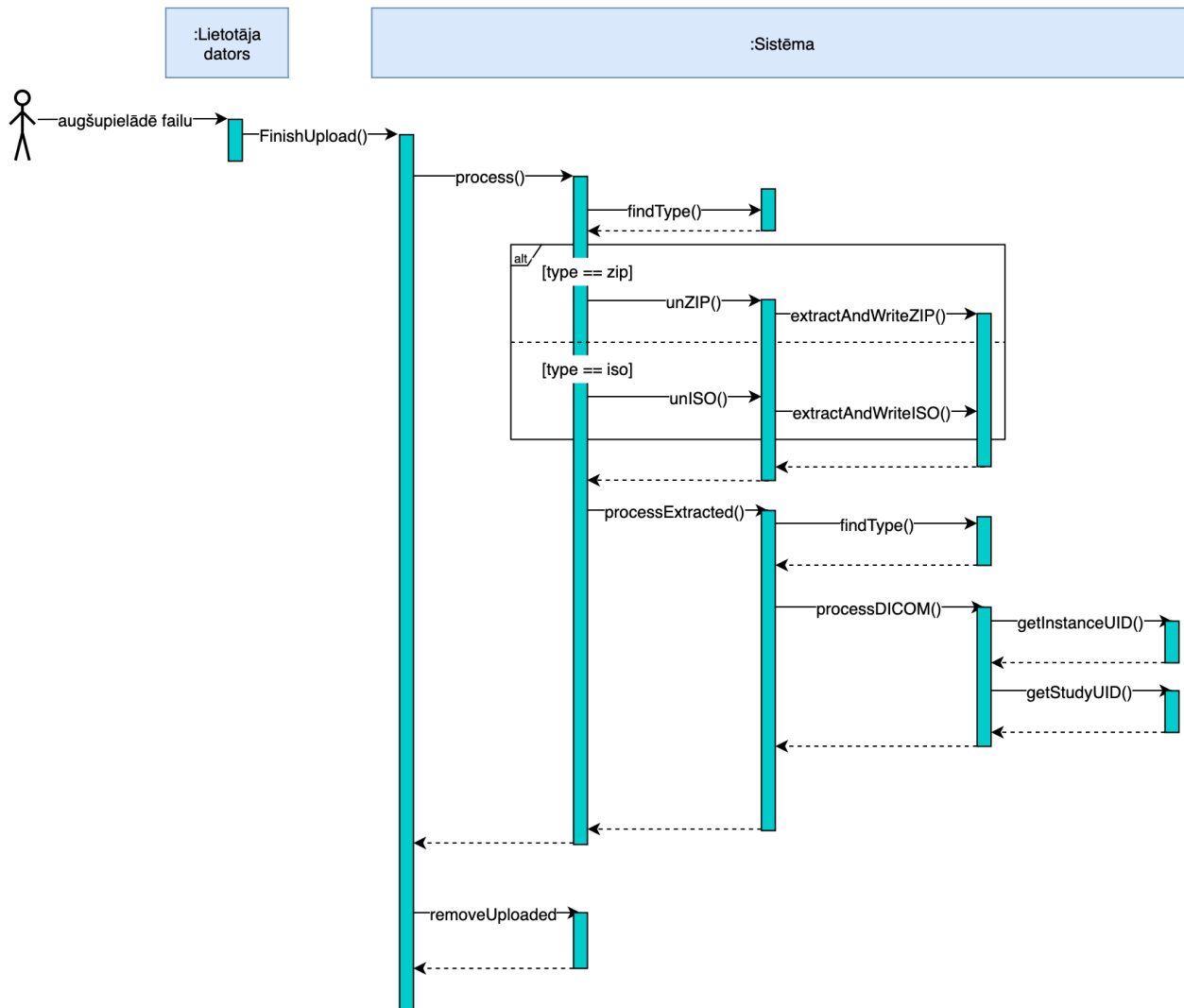
2.4. Daļējs funkciju un lietotāja saskarnes projektējums

2.4.1. Aktivitāšu diagramma



2.3.att. Aktivitāšu diagramma

2.4.2. Viena faila augšupielādes secību diagramma



2.4.att. Secību diagramma

2.4.3. Lietotāja saskarne



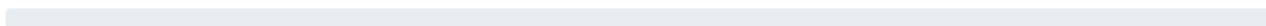
MEDICĪNISKĀS DIAGNOSTIKAS
INFORMĀCIJAS SISTĒMA

IZIET

IZMEKLĒJUMU AUGŠUPIELĀDĒŠANA

Izvēlēties failu

PROGRESS:



Sākt augšupielādi

Veiksmīgi veiktas augšuplādes tiks parādītas šeit:

©DataMed

[Par DataMed](#) | [Kontakti](#) | [Lietošanas noteikumi](#) | [Monitora kalibrēšana](#) | [jautājumi un priekšlikumi](#)

2.5.att. Lietotāja saskarne

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Testēšanas apraksts

Testēšana ir saistīta ar kļūdu meklēšanu sistēmā. Tā kā sistēmai ir izstrādāta programmatūras prasību specifikācija un projektējums, tad testēšana tiks veidota, balstoties uz sistēmas funkcionalitātei izvirzītajām prasībām.

Tika pārbaudīta sistēmas darbība gan ideālos apstākļos, kad visi padotie dati ir valīdi, gan kļūdas apstākļos, lai noskaidrotu, vai sistēma varēs atrast un pareizi identificēt radušos kļūdu. Ja sistēma atgriež kļūdu, tas nenozīmē, ka sistēma darbojas nepareizi - gluži pretēji, jo sistēmai padoto failu saturs ir atkarīgs no paša lietotāja, tāpēc sistēmai ir jāprot identificēt kļūdainos failus un atbilstoši tam reaģēt.

Testa dokumentācija satur laukus ar šādu informāciju: testa numurs, testējamā funkcija, testa apraksts, sagaidāmais rezultāts, testa rezultāts.

3.2. Testpiemēri

3.2.1.tabula Augšupielādes un apstrādes moduļa testēšana

Nr.	Testējamā funkcija	Testa apraksts	Sagaidāmais rezultāts	Testa rezultāts
1.	tusdServer	Neautenticējies lietotājs nosūta POST pieprasījumu, lai augšupielādētu failu.	Augšupielāde nenotiek un lietotājs tiek novirzīts uz pieslēgšanās lapu.	+
2.	tusdServer	Augšupielāde tiek apturēta pusceļā, un lietotājs atslēdzas. Pēc 30 min lietotājs pieslēdzas un mēģina augšupielādēt to pašu failu.	Augšupielāde turpinās no pārtraukuma brīža.	+
3.	leftOverCheck	Sistēma tiek ieslēgta, kad augšupielāžu mapē atrodas līdz galam neaugšupielādēts fails.	Funkcija noskaidros, ka fails nav līdz galam augšupielādēts, tāpēc to izlaidīs.	+
4.	makeDirs	Sistēma tiek ieslēgta, kad direktorijas jau ir	Jaunas direktorijas netiks izveidotas.	+

		izveidotas.		
5.	process	Tiek augšupielādēts fails, kurš sevī satur vairākas studijas.	Katrai studijai tiek izveidota sava direktorijs, kur atrodas katrai studijai piederošās bildes.	+
6.	processDICOM	Tiek augšupielādēta koruptēta studija (bez studijas metadatiem).	Funkcijai neizdodas noskaidrot vajadzīgos metadatus, tāpēc tā atgriež kļūdas paziņojumu. Augšupielādētais fails tiek pārvietots uz kļūdu direktoriju.	+
7.	readConfig	Konfigurācijas struktūra pirmkodā atšķiras no struktūras konfigurācijas failā.	Tiek atgriezts kritiskas kļūdas paziņojums. Sistēma pārtrauc darbību.	+
8.	process	Tiek augšupielādēts fails, kas satur ~ 500 bildes dažādās studijās.	Apstrāde notiek bez kļūdām.	+

3.2.2. tabula Importēšanas moduļa testēšana

Nr.	Testējamā funkcija	Testa apraksts	Sagaidāmais rezultāts	Testa rezultāts
1.	main	Studiju direktorijs ir tukša.	Sistēmas tālākā darbība nenotiek.	+
2.	processStudy	Funkcijai tiek padota vērtība, kura neatbilst DICOM standarta prasībām.	Visa studija tiek pārvietota uz kļūdu direktoriju.	+
3.	processStudy	Funkcijai tiek padota studija, kurai dažas bildes jau eksistē sistēmā.	Funkcija noskaidro, ka studija jau eksistē sistēmā, tāpēc pārbaudīs katru bildi atsevišķi, un, ja tā neeksistē, tad tā tiek importēta sistēmā.	+

3.3. Rezultātu kopsavilkums

Testēšanas laikā tika atrastas dažas kļūdas, kuras tika novērstas. Tāpat tika identificētas potenciālas neapstrādātas kļūdu situācijas, kuras tika novērstas, pamainot koda loģiku. Tika veikta atkārtota testēšana, lai pārliecinātos, ka iepriekš atrastās kļūdas vairs neatkārtojas un ka nav parādījušās jaunas.

4. PROJEKTA ORGANIZĀCIJA

Sistēmu tika plānots izstrādāt, balstoties uz ūdenskrituma modeļa izstrādes procesu. Izstrādes procesa laikā radās jaunas prasības, tāpēc pieturēties pie ūdenskrituma modeļa neizdevās.

Pirms sistēmas programmēšanas sākuma tika izveidota dokumentācija. Jāatzīst, ka dokumentācija ir ļoti mainījusies izstrādes laikā. Pirms izstrādes sākuma tika uzstādīta izstrādei nepieciešamā vide. Tika izmantots SublimeText 3 teksta redaktors ar vairākiem papildinājumiem, lai izstrāde varētu noritēt bez sarežģījumiem.

Sākot darbu pie šīs sistēmas, autoram ir bijusi pieredze HTML, CSS, JavaScript valodās, taču nebija pieredzes Go programmēšanas valodā. Pieredzes trūkuma dēļ daudzas lietas prasīja vairāk laika, nekā bija paredzēts. Ar pašreizējo pieredzi sistēmu varētu izveidot daudz ātrāk. Izstrādes laikā tika apgūta Go programmēšanas valoda, kā arī attīstītas zināšanas HTML, CSS, JavaScript valodās un pat medicīnas jomā. Vēl tika apgūts DICOM standarts.

Sāukumā tika izstrādāta augšupielādes un apstrādes moduļa funkcionalitāte, kas ļāva augšupielādēt un apstrādāt failus. Pēc tam tika izstrādāta importēšanas moduļa funkcionalitāte, kas ļāva importēt apstrādātos studiju failus uzņēmuma PACS serverī.

Sistēmas testēšana notika gan izstrādes laikā, gan pēc izstrādes pabeigšanas. Katra jaunā funkcija tika vairākkārt notestēta, lai pārliecinātos, ka tā tiešām izpilda savu mērķi un ka tā nelabvēlīgi neietekmē citas funkcijas.

Sistēmas izstrādē arī palīdzēja prakses vadītājs, kurš sniedza atbildes uz jautājumiem, palīdzēja ievērot labo praksi programmējot un pozitīvi atbalstīja.

Versijas kontrolei tika izmantots Git, un sistēmas kods kādu laiku tika glabāts Github privātā repozitorijā, taču pēc darba līguma noslēgšanas kods tika noņemts no Github un pārlikts uz uzņēmuma privātā Git servera.

5. KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu izstrādātās sistēmas kvalitāti, tika ievēroti vairāki noteikumi:

- rīka prasību specifikācija un projektējuma apraksts tika noformēti atbilstoši valsts standartiem;
- sistēmas pirmkodā tika ievērots vienots programmēšanas stils, ievērojot Latvijas Universitātē izstrādātos ieteikumus par labo programmēšanas stilu;
- sistēmas pirmkodam tika pievienoti komentāri vietās, kur ir nepieciešams;
- sistēmas pirmkods tika sadalīts vairākos failos, lai uzlabotu pārlasāmību;
- izstrādātā sistēma tika testēta atbilstoši izstrādātai testēšanas dokumentācijai.

6. KONFIGURĀCIJAS PĀRVALDĪBA

Izstrādātās sistēmas konfigurācijas pārvaldībai tika izmantots Git versiju kontroles rīks. Sistēmas kods sākotnēji glabājās Github repozitorijā, taču pēc darba līguma noslēgšanas, kodu nācās noņemt no Github un pievienot uzņēmuma privātajam Git serverim.

Veicot izmaiņas sistēmas kodā, tika pievienoti īsi, bet izsmeļoši komentāri par veiktajām izmaiņām kopš pēdējās saglabāšanas reizes.

7. DARBIETILPĪBAS NOVĒRTĒŠANA

Darba autoram pirms sistēmas izstrādāšanas nebija iepriekšējas pieredzes Go programmēšanas valodā, tāpēc novērtēt sistēmas darbietilpību bija grūti. Pēc prakses vadītāja ieteikuma tika izvēlēts projekts, kurš palīdzētu apgūt jaunas prasmes un tiktu izmantots uzņēmuma produkcijas vidē. Kopumā sistēmas pirmkods sastāvēja no aptuveni 1535 koda rindiņām. Procentuāls sadalījums: Go – 75%, JavaScript – 8%, HTML – 6%, CSS – 11%.

Lai novērtētu darbietilpību, tika izmantota trīskāršā metode: projekts tika sadalīts vairākos izstrādes posmos un katrs posms tika novērtēts no optimistiskā, reālistiskā un pesimistiskā skatu punkta. Tad kopējā darbietilpība tika aprēķināta, izmantojot sekojošo formulu:

$$\frac{S(\text{optimistiskais}) + 4S(\text{reālistiskais}) + S(\text{pesimistiskais})}{6}$$

Par vienu darba dienu tiek uzskatītas 8 stundas.

	Pesimistiskais novērtējums (dienas)	Reālistiskais novērtējums (dienas)	Optimistiskais novērtējums (dienas)
Iepazīšanās ar izmantotajām tehnoloģijām	18	15	10
Programmatūras prasību specifikācijas izveidošana	8	6	4
Programmatūras projektējuma apraksta izveidošana	8	6	4
Konsultācijas ar prakses un darba vadītāju	4	3	2
Augšupielādes un apstrādes moduļa izstrāde	17	15	12
Importēšanas moduļa izstrāde	13	10	7
Testēšanas piemēru izveidošana un testēšana	5	4	2
Testēšanas dokumentācijas izveidošana	4	3	2
Kvalifikācijas darba noformēšana atbilstoši standartiem	7	5	3
Kopā:	84	67	46

Darbietilpība = $(84 + (67 * 4) + 46) / 6 = 66,3$ dienas jeb 3,3 personmēneši.

Pēc formulas redzams, ka darbietilpība ir vienāda ar 3,3 personmēnešiem, kas atbilst kvalifikācijas darba izstrādes prasībām.

SECINĀJUMI

Darba izstrādes laikā tika apgūta Go programmēšanas valoda, attīstītas zināšanas HTML, CSS un JavaScript valodās, apgūts DICOM standarts, kā arī iegūtas jaunas zināšanas medicīnas nozarē. Papildus tam vēl tika apgūts versiju kontroles rīks Git.

Tā kā pieredze ar Go programmēšanas valodu iepriekš nav bijusi, liela daļa laika tika patērēta sintakses apguvei, valodas dokumentācijas lasīšanai. Lielas problēmas sagādāja izstrādes vides sagatavošana, tādēļ ka lielākā daļa uzņēmuma darbinieku izmanto Linux veida operētājsistēmas, taču darba autors izmanto Windows operētājsistēmu. Vēlāk operētājsistēmu atšķirība radīja arī citas problēmas, jo daži rīki, kas bija pieejami uz Linux, nebija pieejami uz Windows.

Kopumā gūto pieredzi, izstrādājot šo sistēmu, var vērtēt kā ļoti pozitīvu, jo izstrādes laikā tika papildinātas teorētiskās zināšanas, attīstītas programmēšanas prasmes un iemaņas, kā arī veidojās interese par plašākām IT izmantošanas iespējām medicīnas nozarē.

IZMANTOTIE AVOTI

- [1] LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis
- [2] LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai
- [3] Alan A. A. Donovan, Brian W. Kernighan: The Go Programming Language, ISBN-13: 978-0134190440 (Pēdējo reizi skatīts 23.05.2019)
- [4] <https://tour.golang.org/> (Pēdējo reizi skatīts 26.05.2019)

PIELIKUMI

1. Pielikums. Augšupielādes un apstrādes moduļa pirmkoda fragmenti

Inicializācijas un galvenās funkcijas realizācija:

```
func init() {
    readConfig(Asset, &conf)
    var err error

    if conf.Name == `` || conf.UploadDir == `` || conf.DicomDir == `` || conf.BadDir == `` || conf.Listen == `` {
        fatal(`bad ConfigData`)
    }

    c := cron.New()
    err = c.AddFunc(conf.NotifySchedule, notify)
    if err != nil {
        fatal(`bad NotifySchedule:`, err)
    }

    c.Start()
    err = makeDirs()
    if err != nil {
        fatal(`failed to make directories`)
    }
}

func main() {
    //check uploads and process fully uploaded ones
    go leftOverCheck()

    store := filestore.FileStore{
        Path: conf.UploadDir, //path where to store uploaded files
    }

    composer := tusd.NewStoreComposer()
    composer.UseFinisher(&Finisher{})
    store.UseIn(composer)

    //starts tusd server that handles file uploading
    go tusdServer(composer)

    r := mux.NewRouter()

    // This will serve static files under :XXXX/a/<filename>
    r.PathPrefix("/a/").Handler(http.StripPrefix("/a/", http.FileServer(assetFS()))))
    r.HandleFunc("/", handleIndex)

    srv := &http.Server{
        Handler: r,
        Addr:    conf.Listen,
    }

    //starts the server
    fatal(srv.ListenAndServe())
}
```

Failu apstrādes funkcijas realizācija:

```
func process(path string) error {
    today := dm.Today().Format(`2006-01-02`)
    fileType, err := findType(path + `.bin`)
    if err != nil {
        return err
    }

    filename := filepath.Base(path) //get the file name (example: "/uploads/151av3as" to "151av3as")
    destPath := filepath.Join(conf.UnzipDir, filename)

    switch fileType {
    case `zip`:
        err = unZIP(path+".bin", destPath)
        if err != nil {
            failed[today]++
            //this removes any extracted files in case unZIP failed midway
            errs := os.RemoveAll(destPath)
            if errs != nil {
                fmt.Println(`unZIP remove error:`, errs)
            }
            return err
        }
        zipCount[today]++
    case `iso`:
        err = unISO(path+".bin", destPath)
        if err != nil {
            failed[today]++
            //this removes any extracted files in case unISO failed midway
            errs := os.RemoveAll(destPath)
            if errs != nil {
                fmt.Println(`unISO remove error:`, errs)
            }
            return err
        }
        isoCount[today]++
    default:
        return errInvalidType
    }

    //processes extracted files
    err = filepath.Walk(destPath, processExtracted)
    if err != nil {
        //this removes any leftover extracted files
        errs := os.RemoveAll(destPath)
        if errs != nil {
            fmt.Println(`processExtracted remove error:`, errs)
        }
    }

    return err
}

//this removes any leftover extracted files
err = os.RemoveAll(destPath)
if err != nil {
    return err
}

return nil
}
```

2. Pielikums. Importēšanas moduļa pirmkoda fragmenti

Inicializācijas un galvenās funkcijas realizācija:

```
func init() {
    readconfig(Asset, &conf)
    var err error

    if conf.Name == `` || conf.DcmDir == `` || conf.BadDir == `` || conf.CompleteDir == `` || conf.Not:
        fatal(`bad ConfigData`)
    }

    conf.runEvery, err = time.ParseDuration(conf.RunEvery)
    if err != nil {
        fatal(`bad RunEvery`)
    }

    c := cron.New()
    err = c.AddFunc(conf.NotifySchedule, notify)
    if err != nil {
        fatal(`bad NotifySchedule:`, err)
    }

    c.Start()

    err = makeDirs()
    if err != nil {
        fatal(`failed to make directories:`, err)
    }
}

func main() {
    for {
        studies, err := ioutil.ReadDir(conf.DcmDir)
        if err != nil {
            fmt.Println(`failed to read DICOM directory:`, err)
        }

        //process every study in DcmDir
        for _, study := range studies {
            path := filepath.Join(conf.DcmDir, study.Name())
            err = processStudy(path)
            if err != nil {
                failedCount[today().Format(`2006-01-02`)]++
                fmt.Println(`failed to process study:`, err, ` (Study: `, study.Name(), `)`)
                err = moveToBadDir(path)
                if err != nil {
                    fmt.Println(`failed to move to badDir:`, err, ` (Study: `, study.Name(), `)`)
                }
                continue
            }
        }

        uploadCount[today().Format(`2006-01-02`)]++
        err = moveToCompleteDir(path)
        if err != nil {
            fmt.Println(`failed to move to completeDir:`, err, ` (Study: `, study.Name(), `)`)
        }
    }

    time.Sleep(conf.runEvery)
}
```

Studijas apstrādes funkcijas realizācija:

```
func processStudy(path string) error {
    files, err := ioutil.ReadDir(path)
    if err != nil {
        return fmt.Errorf("failed to read study: %v", err)
    }

    var uid, institution, patName string
    for i, file := range files {
        t, err := dicom.File(filepath.Join(path, file.Name()), `0020,000D`)
        if err != nil {
            return fmt.Errorf("failed to read (Image: %s):%v", file.Name(), err)
        }
        if i == 0 {
            uid = t.String(`0020,000D`)
            institution = t.String(`0008,0080`)
            patName = t.String(`0010,0020`)
        }
        if uid == `` {
            return fmt.Errorf("invalid uid: (Image: %s)", file.Name())
        }
        if institution == `` {
            return fmt.Errorf("invalid institution name: (Image: %s)", file.Name())
        }
        if patName == `` {
            return fmt.Errorf("invalid patient name:(Image: %s)", file.Name())
        }
        if (institution != t.String(`0008,0080`)) || (patName != t.String(`0010,0020`)) || (uid != t.String(`0020,000D`))
            return fmt.Errorf("wrong institution/patientname/studyuid or uid not real (Image: %s)", t.String(`0008,0018`))
        }
    }
    s, err := findStudy(uid)
    if err != nil {
        return fmt.Errorf("failed to get study: %v", err)
    }
    //check if study exists in PACS. If not ok- send full study, else- send non existent files one by one
    if len(s) == 0 {
        err = send(path)
        if err != nil {
            return fmt.Errorf("failed to send: %v", err)
        }
        studyNotify(uid)
    } else {
        for _, file := range files {
            isPresent := false

            f, err := dicom.File(filepath.Join(path, file.Name()), `0020,000D`)
            if err != nil {
                return fmt.Errorf("failed to read (Image: %s): %v", file.Name(), err)
            }
            for _, k := range s {
                if k.String(`0008,0018`) == f.String(`0008,0018`) {
                    isPresent = true
                    break
                }
            }
            if !isPresent {
                err := send(filepath.Join(path, file.Name()))
                if err != nil {
                    return fmt.Errorf("failed to send: %v", err)
                }
            }
        }
        studyNotify(uid)
    }
    return nil
}
```

Dažādu importēšanas moduļa funkciju realizācijas:

```
func isValidUID(studyUID string) bool {
    if (len(studyUID) < 20) || (strings.Count(studyUID, '.') < 3) || (len(studyUID) > 64) {
        return false
    }
    return true
}

func send(path string) error {
    dati, err := dicom.ParseDcmURI(conf.PACS)
    if err != nil {
        return fmt.Errorf(`failed to parse conf.PACS: %v`, err)
    }

    //if path is directory then adds `+sd` to send multiple files
    if info, err := os.Stat(path); err == nil && info.IsDir() {
        errMessage, err := localExec(`storeescu`, `-aet`, dati.Client, `-aec`, dati.Server, dati.IP, dati.Port, path, `+sd`)
        if err != nil {
            return fmt.Errorf(errMessage, err)
        }
    } else {
        errMessage, err := localExec(`storeescu`, `-aet`, dati.Client, `-aec`, dati.Server, dati.IP, dati.Port, path)
        if err != nil {
            return fmt.Errorf(errMessage, err)
        }
    }
    return nil
}

func moveToBadDir(path string) error {
    badDirName := filepath.Base(path)
    badDirPath := filepath.Join(conf.BadDir, badDirName)
    return os.Rename(path, badDirPath)
}

func moveToCompleteDir(path string) error {
    completeDirName := filepath.Base(path)
    completeDirPath := filepath.Join(conf.CompleteDir, completeDirName)
    return os.Rename(path, completeDirPath)
}

func makeDirs() error {
    err := os.MkdirAll(conf.CompleteDir, 0755)
    if err != nil {
        return err
    }
    err = os.MkdirAll(conf.BadDir, 0755)
    if err != nil {
        return err
    }
    return nil
}

func studyNotify(studyUID string) {
    if !conf.StudyNotify {
        return
    }
    date := today().Format(`2006-01-02`)
    s := `[` + date + `] Files from study (` + studyUID + `) have been uploaded`

    err := dm.Notify(conf.Name, s)
    if err != nil {
        fmt.Println(err)
    }
}
```

Kvalifikācijas darbs „**Failu augšupielādes un apstrādes sistēma IT uzņēmuma**” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: **Linards Losāns** _____ **.05.2019.**

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: **Dr.sc.comp Māris Vītiņš** _____ **.05.2019.**

Recenzents: **Agnis Škuškovniks**

Darbs iesniegts 27.05.2019.

Kvalifikācijas darbu pārbaudījumu komisijas sekretāre: *Darja Solodovņikova* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____