

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

INKASĀCIJAS SISTĒMAS UZLABOŠANA

KVALIFIKĀCIJAS DARBS

Autors: Toms Mucenieks

Studenta apliecības Nr.: tm13023

Darba vadītājs: Ms. Sc. Comp. Kalvis Upītis

RĪGA 2017

ANOTĀCIJA

Kvalifikācijas darba ietvaros tika veikta inkasācijas sistēmas pārrakstīšana, saglabājot esošo funkcionalitāti. Sistēma sastāv no 2 komponentēm: servera puses lietotne, kur tiek glabāta visu inkasentu dati un *Windows CE* portatīvā datu termināļa puses lietotne, kurā inkasents ievada datus un sinhronizē tos ar serveri. Termināļa puses lietotne, izmantojot *React* izstrādes ietvaru, tika pārrakstīta mobilajām ierīcēm. Šī darba mērķis ir:

pārrakstīt esošo termināļa lietotni mobilajām ierīcēm (*Android, iOS*)

atjaunot novecojušu programmatūru

ieekonomēt līdzekļus, inkasācijai izmantojot darbiniekiem izdotos viedtālruņus

Jaunā lietotne viedtālruņos nodrošinās iespēju sinhronizēt datus ar serveri, skenēt svītrkodus, izmantojot tālruņa kameru, kā arī inkasenti spēs tajā ievadīt nepieciešamo informāciju atskaitēm.

Atslēgvārdi: viedtālrunis, inkasācijas sistēma, *Windows CE, React, Android, iOS*

ABSTRACT

ENCASHMENT SYSTEM IMPROVEMENT

Within the scope of qualification work author rewrote encashment system, maintaining the same functionality. The system consists of 2 components: a server-side app for storing all data related to encashments and *Windows CE* portable data terminal app where the encashment person enters data and synchronizes device to the server. Terminal app was rewritten to the mobile devices by using React framework. The goal of this work is to:

rewrite existing terminal app to mobile devices (*Android, iOS*)

upgrade outdated software

save money by using the company's issued smartphones

The new app for smartphones will make it possible to synchronize data with the server, to scan barcodes using phone's camera as well as people who do encashments will be able to enter the necessary information for reports.

Keywords: smartphone, encashment system, *Windows CE, React, Android*

SATURS

IEVADS	7
VĀRDNĪCA.....	9
1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA	11
1.1. Ievads.....	11
1.1.1. Nolūks	11
1.1.2. Darbības sfēra	11
1.1.3. Saistība ar citiem dokumentiem	11
1.1.4. Pārskats	11
1.2. Vispārējs apraksts	11
1.2.1. Sistēmas lietotāji.....	11
1.2.2. Lietotāja raksturierzīmes	11
1.2.3. Produkta funkcijas	12
1.2.4. Vispārējie ierobežojumi	12
1.3. Funkcionālās prasības – Lietotāju stāsti.....	12
1.3.1. Autentifikācijas dati	12
1.3.2. Datu sinhronizācija	13
1.3.3. Inkasācija.....	13
1.3.3.1. Inkasācijas panelis.....	14
1.3.3.2. Adrešu saraksts	14
1.3.3.3. Spēļu automātu saraksts	15
1.3.3.4. Inkasācijas datu ievade.....	16
1.3.3.5. Inkasācijas papilddatu ievade	17
1.3.3.6. Inkasācijas aktu drukāšana	17
1.3.3.7. Inkasācijas aktu numuru saraksts.....	18

1.3.3.8.	Inkasācijas akta numura anulēšana	18
1.3.3.9.	Inkasācijas numura piesaiste aktam	19
1.4.	Nefunkcionālās prasības	20
1.4.1.	Tīkla pieejamība	20
1.4.2.	Atbalstītās versijas	20
1.4.3.	Drošība	20
2.	PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS	21
2.1.	Ievads.....	21
2.1.1.	Nolūks	21
2.1.2.	Darbības sfēra	21
2.1.3.	Saistība ar citiem dokumentiem	21
2.1.4.	Pārskats	21
2.2.	Datu plūsmu diagrammas	22
2.2.1.	0. līmenis	24
2.2.2.	1. līmenis – inkasācijas sistēma	25
2.2.3.	2. līmenis – viedtālrunu lietotne	26
2.2.4.	3. līmenis – datu sinhronizācija	28
2.2.5.	3. līmenis – inkasācijas datu pārvaldība.....	30
2.2.6.	3. līmenis – inkasācijas aktu pārvaldība.....	33
2.3.	Datubāzes diagrammas.....	35
2.3.1.	Konceptuālais ER modelis	35
2.3.2.	Fiziskais ER modelis	38
2.3.3.	Datubāzes tabulu apraksts.....	38
2.4.	Moduļu projektējums	57
2.4.1.	Datu sinhronizācija.....	57
2.4.2.	Inkasācijas datu pārvaldība	58

2.4.3. Inkasācijas aktu pārvaldība	59
2.5. Lietotāja ekrāna formu projektējums	60
2.5.1. Inkasācijas panelis	61
2.5.2. Datu sinhronizācija	62
2.5.3. Sinhronizējamo datu labošana	63
2.5.4. Autentifikācijas dati.....	64
2.5.5. Adresez izvēle	65
2.5.6. Adresez informācija.....	66
2.5.7. Instalāciju izvēle.....	67
2.5.8. Instalācijas informācija	69
2.5.9. Inkasācijas datu ievade	70
2.5.9.1. Skaitītāju dati	71
2.5.9.2. Naudas uzpilde.....	73
2.5.10. IA numuru saraksts.....	74
2.5.11. IA numura pievienošana	75
2.5.12. IA numura anulēšana	76
2.5.13. Par lietotni	77
3. TESTĒŠANAS DOKUMENTĀCIJA.....	78
3.1. Ievads.....	78
3.2. Testējamās raksturierzīmes	78
3.3. Testpiemēru specifikācija	78
3.3.1. Funkcionālās prasības	79
3.3.1.1. Datu sinhronizācija.....	79
3.3.1.2. Inkasācijas datu pārvaldība.....	80
3.3.1.3. Inkasācijas aktu pārvaldība.....	83
3.3.2. Nefunkcionālās prasības.....	85

4.	PROJEKTA ORGANIZĀCIJA	86
5.	KVALITĀTES NODROŠINĀŠANA.....	87
6.	KONFIGURĀCIJU PĀRVALDĪBA	88
7.	DARBIETILPĪBAS NOVĒRTEJUMS.....	89
8.	PIELIKUMI	90
8.1.	Inkasācijas akta izdrukas paraugs	90
8.2.	Automatizēto vienībtestu rezultātu piemēri.....	91
8.3.	Konfigurācijas pārvaldības	92
8.3.1.	Git	92
8.3.2.	Jenkins.....	92
8.3.3.	Fabric	93
8.4.	Koda fragmenti	94
8.4.1.	Datu sinhronizācijas skats	94
8.4.2.	XML formāts datu nosūtīšanai uz serveri	96
9.	SECINĀJUMI.....	98
	LITERATŪRAS SARAKSTS	99

IEVADS

Uzņēmumam “Joker” pieder plašs spēļu zāļu tīkls visā Latvijā - tas tiek pārstāvēts 28 pilsētās un nodrošina 60 spēļu zāļu darbību. Uzņēmums izmanto inkasācijas sistēmu naudas ieņēmumu uzskaitēi no spēļu automātiem, ko ir izstrādājis SIA Galeo Consulting. Sistēmas lietotāji ir uzņēmuma “Joker” spēļu zāļu inkasenti, kā arī grāmatveži. Sistēma sastāv no tīmekļa lietotnes (servera), kas paredzēts lietotāju administrēšanai un inkasācijas datu pārskatu veidošanai, kā arī no portatīvā datu termināļa lietotnes, kas tiek izmantota inkasācijas veikšanai un automātu labošanas datu ievadei, taču tai piemīt sekojoši trūkumi:

- Lietotne darbojas uz novecojušas operētājsistēmas (*Windows CE 5.0*), kas vairs netiek oficiāli atbalstīta kopš 2014. gada.[2]
- Ierobežota pieeja tīklam (pieejams tikai *WiFi*).
- Inkasentiem un mehāniķiem nepieciešams nēsāt līdzī portatīvo datu termināli, kā vietā viņi varētu izmantot inkasācijas veikšanai un automātu labošanai uzņēmuma sagādātos darba telefonus.

Lai šos trūkumus novērstu, kvalifikācijas darba autors ir pārrakstījis esošo termināļa lietotni priekš *Android* un *iOS* viedtālruniem, kas sīkāk tiek aprakstīts turpmākajās nodaļās (tā kā kvalifikācijas darba apjomam ir jāatbilst 3 personmēnešiem, kvalifikācijas darba ietvaros tiek pārrakstīta tikai tā lietotnes daļa, kas paredzēta inkasentiem).

Darba mērķis ir izveidot spēļu automātu inkasācijas sistēmas lietotni, kas aizvietos portatīvo datu termināli. Šī mērķa sasniegšanai tika izstrādāta šāda funkcionalitāte, izmantojot sekojošas tehnoloģijas:

- Inkasācijas datu ievade un attēlošana, ko realizē vienas tīmekļa lapas lietotne viedtālruniem. Lapa veidota, izmantojot *JavaScript* bibliotēku – *ReactJS*.
- Lokāla datubāze - tiek glabāti dati sinhronizācijai ar serveri. Datubāze tiek veidota, izmantojot *IndexedDB* saskarni un *Lovefield* ietvaru relāciju simulēšanai.
- *React-Native* radīts tīmekļa pārlūkprogrammas skats, kurā tiek ievietota *ReactJS* lietotne.
- Svītrkodu skenēšana – spēļu automātu svītrkodu skenēšanai, realizēts izmantojot *React-Native* ietvaru.
- Datu sinhronizācija – izmantojot *Axios* bibliotēku datu apmaiņai ar serveri.

Kvalifikācijas darba ietvaros izstrādātā lietotne ļaus, izmantojot telefona kameru, noskenēt svītrkodu ātrai spēļu automāta lokalizēšanai, ievadīt nepieciešamo informāciju par konkrēto automātu un sinhronizēt datus ar serveri, kā arī atcelt vajadzību pēc portatīvā datu termināļa.

VĀRDNĪCA

Jēdziens	Skaidrojums
API	Saskarne, kuru datora lietojumprogramma var izmantot, lai piekļūtu trešo pušu funkcijām vai programmām.
Android	Mobilā operētājsistēma, kas balstās uz atvērtā koda principa.
JavaScript	Skriptu valoda, kura tiek izmantota, lai veidotu interaktīvu saturu priekš tīmekļa lietotnēm.
Git	Programmatūras versiju kontroles sistēma.
GitLab	Programmatūras versiju kontroles sistēmas mājaslapa.
HTTP	Lietotņu protokols, kas nosaka, kā dati tiks pārsūtīti caur internetu.
HTTPS	Lietotņu protokols, kas nosaka, kā dati tiks pārsūtīti caur internetu drošā veidā.
iOS	Mobilā operētājsistēma, kas paredzēta tikai priekš kompānijas “Apple” izstrādātajiem produktiem.
Jenkins	Atvērtā pirmkoda automatizācijas serveris, kas tiek izmantots programmatūras būvēšanai, testēšanai un izvietošanai.[9]
Jest	Javascript ietvars automatizētai vienībtestēšanai.
POST	HTTP pieprasījuma metode datu nosūtīšanai uz tīmekļa serveri
IndexedDB	JavaScript API, ar kura palīdzību var izveidot objekt-orientētu datubāzi.
LoveField	Javascript ietvars darbam ar IndexedDB.
ReactJS	Javascript bibliotēka, kas ļauj ērti un viegli veidot un uzturēt skatus iekš interneta pārlūkprogrammas.
React-Native	Ietvars, kas ļauj izveidot lietotnes gan uz Android, gan uz iOS mobilajām ierīcēm, izmantojot JavaScript.
Redmine	Elastīga projektu pārvaldības sistēma.[3]
SCRUM	Iteratīvs spējās izstrādes ietvars programmatūras izstrādes pārvaldībai.
SSL	Kriptogrāfisks protokls, kas nodrošina drošu savienojumu caur tīklu.
URL	Vienotais resursu vietradis, kas norāda interneta pārlūkprogrammā, kur atrast konkrēto interneta resursu.
XML	Paplašināmās iezīmēšanas valoda, kas speciāli izstrādāta darbam ar tīmekļa dokumentiem. Valoda XML nodrošina globālā tīmekļa izstrādātajiem iespēju

	radīt savas personiskās birkas (kodus), lai nodrošinātu tādu funkciju izpildi, ko nevar nodrošināt ar valodas HTML starpniecību.
Windows CE	Operētājsistēma, kas veidota priekš mobilām vai cita veida izmēra ierobežotām ierīcēm.
WiFi	Bezvadu lokālais tīkls

1. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

1.1. Ievads

1.1.1. Nolūks

Šajā dokumentā tiek apkopotas funkcionālās prasības lietotāju stāstu formā un nefunkcionālās prasības.

1.1.2. Darbības sfēra

Izstrādātā aplikācija paredzēta “Joker” spēļu zāļu automātu inkasācijai.

1.1.3. Saistība ar citiem dokumentiem

Dokuments saistīts ar programmatūras projektējuma aprakstu un testēšanas dokumentāciju.

1.1.4. Pārskats

Programmatūras projektējuma apraksts sastāv no 4 daļām:

- Ievads – informācija, kas satur dokumenta nolūku, darbības sfēru, kā arī norāda uz saistību ar citiem dokumentiem.
- Vispārējs apraksts – sniedz īsumā informāciju par lietotnes funkcijām, lietotājiem, kā arī tiek aprakstīti pieņēmumi, atkarības un sistēmas ierobežojumi.
- Funkcionālās prasības – informācija par visām lietotnes funkcionālām prasībām.
- Nefunkcionālās prasības – informācija par visām lietotnes nefunkcionālām prasībām.

1.2. Vispārējs apraksts

1.2.1. Sistēmas lietotāji

Tā kā šī lietotne (kvalifikācijas darba ietvaros) ir paredzēta tikai inkasentiem, ir tikai viena lietotāju grupa:

- Inkasents – Inkasācijas sistēmas datubāzē reģistrēts lietotājs, kam ir pieejamas visas funkcijas, ko piedāvā šī lietotne.

1.2.2. Lietotāja raksturiezīmes

Ar lietotni var strādāt tikai un vienīgi “Joker” kompānijas apmācīts spēļu automātu inkasents.

1.2.3. Produkta funkcijas

Inkasenta funkcijas:

- Pievienot inkasācijas ierakstus
- Dzēst un rediģēt inkasenta ievadītos, bet nesinhronizētos inkasācijas ierakstus
- Izdrukāt inkasācijas aktus
- Sinhronizēt inkasācijas datus ar serveri

1.2.4. Vispārējie ierobežojumi

Darbam ar lietotni ir nepieciešams Android viedtālrunis, kam operētājsistēmas versija ir vismaz 4.4 (KitKat) vai iOS viedtālrunis, kam operētājsistēmas versija ir vismaz 8.0.

1.3. Funkcionālās prasības – Lietotāju stāsti

Šajā nodaļā tiek aprakstītas programmatūras funkcionālās prasības, kurām ir jāatbilst izveidotajai lietotnei. Prasības tiek aprakstītas pēc spējas izstrādes lietotāju stāstu veidā un uzdotas Redmine pārvaldības rīkā, lietotāju stāstu identifikatori sakrīt ar Redmine automātiski piešķirtajiem identifikatoriem. Katra prasība tiek attēlota tabulā ar šādām sadaļām: stāsts, akceptēšanas kritērijs un Redmine identifikators.

1.3.1. Autentifikācijas dati

Stāsts
Es kā inkasents vēlos, lai man būtu iespēja ievadīt savus autentifikācijas datus un tie tiktu saglabāti, lai man vairākas reizes nebūtu jāvada no jauna.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Skatā jābūt šādiem obligātajiem laukiem:<ul style="list-style-type: none">○ lietotājvārds,○ parole.• Ja lietotājs jau ir ievadījis autentifikācijas datus, tad tiem jāparādās attiecīgajos laukos, taču parolei jābūt aizzīmētai.• Ja lietotājs vēlreiz saglabā datus, tad saglabātie dati tiek izmainīti uz tikko ievadītajiem.
Identifikators
1897

1.3.2. Datu sinhronizācija

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams izlabot nekorekti ievadītos datus un sinhronizēt datus ar serveri, lai varu aizsūtīt ievadītos datus uz serveri un saņemt jaunus no servera.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Ja lietotājs ir kļūdaini ievadījis lietotājvārdu un/vai paroli, tad sinhronizācijas vietā tiek atgriezts paziņojums, kas apraksta kļūdas iemeslu.• Ja lietotājs ir pareizi ievadījis autentifikācijas datus, bet vēl nav veicis datu izmaiņas, tad notiek tikai datu lejupielāde par konkrēto lietotāju no servera.• Ja lietotājs ir pareizi ievadījis autentifikācijas datus, bet ir jau sinhronizējis, tad sākumā notiek datu augšupielāde serverī, un tikai tad tiek lejupielādēti dati par konkrēto lietotāju.• Datu augšupielādei un lejupielādei ir jābūt atomārai – veiksmīgi tiek sinhronizēti visi dati vai neviens no tiem.• Ja savienojums ir neveiksmīgs, tad sinhronizācijas vietā tiek atgriezts paziņojums, kas apraksta kļūdas iemeslu.• Ja, sinhronizējot datus ar serveri, tiek konstatēts, ka ievadītie inkasācijas dati ir kļūdaini, tad lietotājam tiek dota iespēja izdzēst nepareizi ievadītos datus.
Identifikators
1898

1.3.3. Inkasācija

Visi lietotāju stāsti, kas aprakstīti šajā apakšnodaļā definē to, kas nepieciešams, lai veiktu inkasācijas datu ievadi. Lai būtu iespējams izpildīt jebkuru no šīm prasībām, lietotājam ir jābūt sinhronizētam, ko veic prasība - **1898 Datu sinhronizācija**.

1.3.3.1. Inkasācijas panelis

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams redzēt man piešķirto spēļu automātu skaitu un vietu skaitu, kur šie spēļu automāti atrodas, kā arī pogu uz vietu sarakstu, lai varētu redzēt, cik man ir piešķirti automāti, un uz cik vietām man ir jābrauc, kā arī pāriet uz adresu sarakstu.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Ja lietotājs nav sinhronizējis datus, tad inkasācijas paneļa vietā tiek atgriezts paziņojums, par datu sinhronizācijas nepieciešamību• Ja lietotājs ir sinhronizējis datus, tad tiek parādīts lietotājam pieejamo adresu un automātu skaits.• Ja lietotājs ir sinhronizējis datus, tad lietotājam ir iespēja pāriet uz nākamo skatu, ko realizē prasība – 1906 Adrešu saraksts.
Identifikators
1905

1.3.3.2. Adrešu saraksts

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams apskatīt man piešķirto adresu sarakstu, kam jābūt sakārtotam pēc alfabēta, kā arī apskatīt papildus informāciju par konkrēto adresi un piekļūt konkrētās adreses spēļu automātu sarakstam, lai varu atlasīt spēļu automātu sarakstu izvēlētai adresei.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Adrešu sarakstam jābūt sakārtotam alfabēta secībā.• Lietotājs var apskatīt papildu informāciju par konkrēto adresi.• Lietotājs var atvērt spēļu automātu sarakstu izvēlētai adresei, ko realizē prasība – 1915 Spēļu automātu saraksts.
Identifikators
1906

1.3.3.3. Spēļu automātu saraksts¹

Stāsts
<p>Es kā inkasents vēlos, lai man būtu iespējams apskatīt man piešķirto spēļu automātu sarakstu konkrētai adresei, kā arī izvēlēties automātu kā manuāli, tā arī skenējot konkrēto svītrkodu, kas ir unikāls katram spēļu automātam, lai varu ātri atlasīt man nepieciešamo spēļu automātu tālākai datu ievadei.</p>
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Ja spēļu automāts ir “Jackpot”, tad tas ir jārāda pirmais sarakstā, un tam jābūt speciāli izceltam.• Automāti, kas nav “Jackpot”, ir jārāda alfabēta secībā zem “Jackpot” automātiem.• Ja svītrkodu lasītājs nevar atrast konkrēto spēļu automātu, tad tiek atgriezts paziņojums, kas apraksta kļūdas iemeslu.• Ja automātam jau ir ievadīti inkasācijas dati, tad to kopsavilkums ir jāparāda spēļu automātu sarakstā - blakus konkrētajam automātam.• Lietotājs var apskatīt papildus informāciju par konkrēto automātu.• Lietotājs var noskenēt instalācijas svītrkodu, kas automātiski atver noskanētās instalācijas inkasācijas datu ievades skatu, ko realizē prasība – 1917 Inkasācijas datu ievade.• Ja, skenējot, instalāciju sarakstā netiek atrasta noskenētā instalācija, tad tiek atgriezts paziņojums, kas apraksta kļūdas iemeslu.• Ja inkasācija ir beigusies, dot lietotājam iespēju pāriet uz drukāšanas skatu un inkasācijas numura ievades skatu.• Ja kaut viens inkasācijas ieraksts ticis izveidots, lietotājs var atvērt inkasācijas drukāšanas skatu, ko realizē prasība – 1918 Inkasācijas aktu drukāšana• Ja kaut viens inkasācijas ieraksts ticis izveidots, lietotājs var atvērt inkasācijas aktu numuru saraksta skatu, ko realizē prasība – 1919 Inkasācijas aktu numuru saraksts• Lietotājs var atvērt izvēlētā automāta inkasācijas datu ievades skatu, ko realizē prasība – 1917 Inkasācijas datu ievade.
Identifikators
1915

¹ Spēļu automāts konkrētajā adresē tiek saukts arī par instalāciju.

1.3.3.4. Inkasācijas datu ievade

Stāsts
<p>Es kā inkasents vēlos, lai man būtu iespējams ievadīt sākuma un beigu naudas vienību skaitu, kas atrodas automāta gan mehāniskajā, gan elektroniskajā skaitītājā, redzēt informāciju par iepriekš ievadītajiem un pašreizējiem datiem, kā arī iespēja izlabot tikko ievadītos rādītājus, lai šos datus varētu izmantot sinhronizācijai ar serveri.</p>
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Skatā jābūt šādiem neobligātajiem laukiem:<ul style="list-style-type: none">○ mehāniskais skaitītājs sākumā,○ mehāniskais skaitītājs beigās,○ elektroniskais skaitītājs sākumā,○ elektroniskais skaitītājs beigās.• Ja kāds no augstāk minētajiem laukiem nav aizpildīts, tad, akceptējot inkasācijas ievadītos datus, lauku vērtība būs "0".• Ja spēļu automāts ir "Jackpot", tad ievadīt var tikai beigu naudas vienību skaitu.• Ja šis skats tiek parādīts pirmo reizi, tad automātiski tiek izveidota deklarācija konkrētajai ielai, kur dotais automāts atrodas.• Ja deklarācija jau tikusi izveidota, tad tiek izmantota esošā.• Ja tiek akceptēti ievades dati, tad inkasācijas ieraksts tiek piesaistīts konkrētai inkasācijas aktam un instalācijai.• Lietotājs var pārslēgties no mehāniskā uz automātisko skaitītāju un pretēji.• Pēc ievadīto datu akceptēšanas, naudas vienību ievade tiek nobloķēta, lai atbloķētu ir jāspiež poga "Labot".• Ja inkasācijas dati jau tikuši ievadīti iepriekš, tad lauki tiek automātiski aizpildīti ar iepriekš ievadītajiem datiem un nobloķēti.• Lietotājs var atvērt izvēlētā automāta uzpildes datu ievades skatu, ko realizē prasība – <p>1917 Inkasācijas papilddatu ievade.</p>
Identifikators
1916

1.3.3.5. Inkasācijas papilddatu ievade

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams ievadīt naudas vienību viltojumus, kas atradās automātā, manis uzpildīto naudas vienību skaitu, manis izņemot naudas vienību daudzumu no automāta, kā arī atzīmēt, ka automātam nepieciešama demontāža, lai varu ievadīt sistēmā datus par automāta uzpildi.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Skatā jābūt šādiem neobligātajiem laukiem:<ul style="list-style-type: none">○ viltojumi○ uzpilde○ samazināšana• Ja kāds no augstāk minētajiem laukiem nav aizpildīts, tad, akceptējot ievadītos datus, lauku vērtība būs “0”.• Ja automāta uzpildes dati jau tikuši ievadīti iepriekš, tad tie tiek automātiski parādīti ievades laukos.• Lietotājs var atzīmēt, vai automātam ir nepieciešama demontāža.
Identifikators
1917

1.3.3.6. Inkasācijas aktu drukāšana

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams izdrukāt manis ievadītos inkasācijas datus konkrētai adresei, lai varu uzlikt zīmogu ar man piesaistītu, unikālu numuru, kas ar katra zīmoga lietošanas reizi mainās, un pievienot šo pašu numuru inkasācijas aktam.
Akceptēšanas kritērijs
<ul style="list-style-type: none">• Ja ar printeri nav iespējams savienoties, tad tiek atgriezts paziņojums, kas apraksta kļūdas iemeslu.• Akta izdrukai jāatbilst likumā noteiktajai struktūrai.[4] Izdrukātajā dokumentā jābūt sekojošai informācijai(sīkāk skat. pielikumā – Inkasācijas akta izdrukas paraugs):

<ul style="list-style-type: none"> ○ Dokumenta galvene satur informāciju par uzņēmumu. ○ Dokumenta ķermenis satur inkasācijas tabulu ar datiem, ko inkasents ir ievadījis. ○ Dokumenta kājene satur inkasācijas datu kopsavilkumu un parakstu joslu. <ul style="list-style-type: none"> ● Pēc drukāšanas tiek automātiski atvērts inkasācijas numuru saraksta skats, ko realizē prasība – 1919 Inkasācijas aktu numuru saraksts.
Identifikators
1918

1.3.3.7. Inkasācijas aktu numuru saraksts

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams redzēt sarakstu ar inkasācijas aktu numuriem, lai varu pārskatīt manis ievadītos inkasācijas aktu numurus un pievienot jaunu numuru vai anulēt kļūdaini izlietoto.
Akceptēšanas kritērijs
<ul style="list-style-type: none"> ● Inkasācijas numuru sarakstam jābūt sakārtotam augošā secībā. ● Inkasācijas aktu numuru sarakstā jābūt iekļautiem gan izlietotajiem, gan anulētajiem inkasācijas numuriem. ● Inkasents var izvēlēties numura anulēšanu, ko realizē prasība – 1920 Inkasācijas akta numura anulēšana. ● Inkasents var izvēlēties pievienot jaunu inkasācijas akta numuru, ko realizē prasība – 1921 Inkasācijas numuru piesaiste aktam
Identifikators
1919

1.3.3.8. Inkasācijas akta numura anulēšana

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams anulēt man piešķirto, kļūdaini izlietoto inkasācijas akta numuru, lai vēlāk varu aplūkot anulētos inkasācijas aktu numurus konkrētajā adresē.
Akceptēšanas kritērijs

<ul style="list-style-type: none"> • Skatā jābūt šādiem obligātajiem laukiem: <ul style="list-style-type: none"> ○ sērija ○ numurs • Skatā jābūt šādam neobligātajam laukam: piezīmes. • Ja kāds no augstāk minētajiem obligātajiem laukiem nav aizpildīts, tad tiek atgriezts paziņojums par kļūdas iemeslu. • Ja ievadītais inkasācijas numurs nesakrīt ar sēriju vai nav piešķirts konkrētajam darbiniekam, tad tiek atgriezts paziņojums par kļūdas iemeslu.
Identifikators
1920

1.3.3.9. Inkasācijas numura piesaiste aktam

Stāsts
Es kā inkasents vēlos, lai man būtu iespējams ievadīt datus par uzspiesto zīmoga numuru uz attiecīgā inkasācijas akta, lai vēlāk varu aplūkot izlietos inkasācijas aktu numurus konkrētajā adresē.
Akceptēšanas kritērijs
<ul style="list-style-type: none"> • Skatā jābūt šādiem neobligātajiem laukiem: <ul style="list-style-type: none"> ○ sērija ○ numurs • Ja kāds no augstāk minētajiem obligātajiem laukiem nav aizpildīts, tad tiek atgriezts paziņojums par kļūdas iemeslu. • Ja ievadītais inkasācijas numurs nesakrīt ar sēriju vai nav piešķirts konkrētajam darbiniekam, tad tiek atgriezts paziņojums par kļūdas iemeslu.
Identifikators
1921

1.4. Nefunkcionālās prasības

1.4.1. Tīkla pieejamība

Lietotnei jāspēj veikt datu sinhronizāciju gan caur Wifi tīklu, gan izmantojot mobilo internetu, taču datu ievade ir jānodrošina bezsaistes režīmā. (Identifikators testēšanai – NP1003)

1.4.2. Atbalstītās versijas

Lietotnei jāatbalsta Android operētājsistēma, sākot no 4.4 versijas un iOS operētājsistēma, sākot no 8.0 versijas. (Identifikators testēšanai – NP1003)

1.4.3. Drošība

Sinhronizācijai ar serveri jāizmanto SSL savienojums, lai datus nevarētu pārtvert un nolasīt trešās puses. (Identifikators testēšanai – NP1003)

2. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

2.1. Ievads

Veidojot programmatūras projektējuma aprakstu, lietotne loģiski tiks sadalīta moduļos. Katrs modulis attieksies uz vienu vai vairākām lietotāju prasībām. Lietotne tiek veidota, kā uzņēmuma “Joker” automātu inkasācijas sistēmas daļa. Kvalifikācijas darba ietvaros veidotā lietotne tiek bāzēta uz *React-Native* ietvara, kurā ir Interneta pārlūkprogrammas skats, kurā tiek ievietota *ReactJS* izstrādātā tīmekļa lapa.

2.1.1. Nolūks

Programmatūras projektējuma apraksts (PPA) ir izstrādāts un paredzēts “Inkasācijas sistēmas uzlabošana” projektējuma aprakstīšanai atbilstoši pēc lietotāju stāstos izvirzītajām prasībām.

2.1.2. Darbības sfēra

Izstrādātā aplikācija paredzēta “Joker” spēļu zāļu automātu inkasācijai.

2.1.3. Saistība ar citiem dokumentiem

Dokumenta noformēšanā ievērotas standarta LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai” prasības.

2.1.4. Pārskats

Programmatūras projektējuma apraksts sastāv no 4 daļām:

- Ievads – informācija, kas satur dokumenta nolūku, darbības sfēru, kā arī norāda uz saistību ar citiem dokumentiem.
- Datu plūsmu diagrammas – sniedz grafisku reprezentāciju par to, kā sistēma sistēmas komponentes komunicē savā starpā.
- Datu bāzes projektējums – grafiski reprezentē un apraksta datu bāzes struktūru.
- Moduļu projektējums – apraksta katru sistēmas moduli.

2.2. Datu plūsmu diagrammas

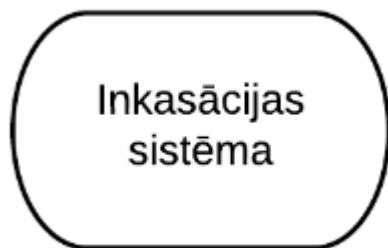
Datu plūsmu diagrammās, galvenokārt, tiek attēlota tā daļa, ko darba autors ir izveidojis vai modificējis. Papildus tam 1. līmeņa datu plūsmu diagrammā tiek attēlots terminālis un serveris ar datubāzēm, lai varētu labāk saprast sistēmas apjomu un datu plūsmas tās moduļiem.

Labākai uzskatāmībai datu plūsmas starp sistēmas moduļiem tiek numurētas. Numerācijas atšifrējumus var atrast zem konkrētās datu plūsmu diagrammas.

Datu plūsmu diagrammu apzīmējumi²:

Apzīmējums

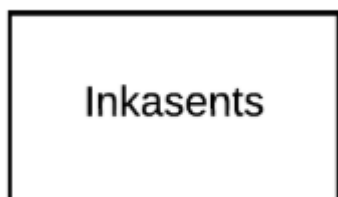
Paskaidrojums



Prasību modulis



Datu plūsma



Lietotāju grupa

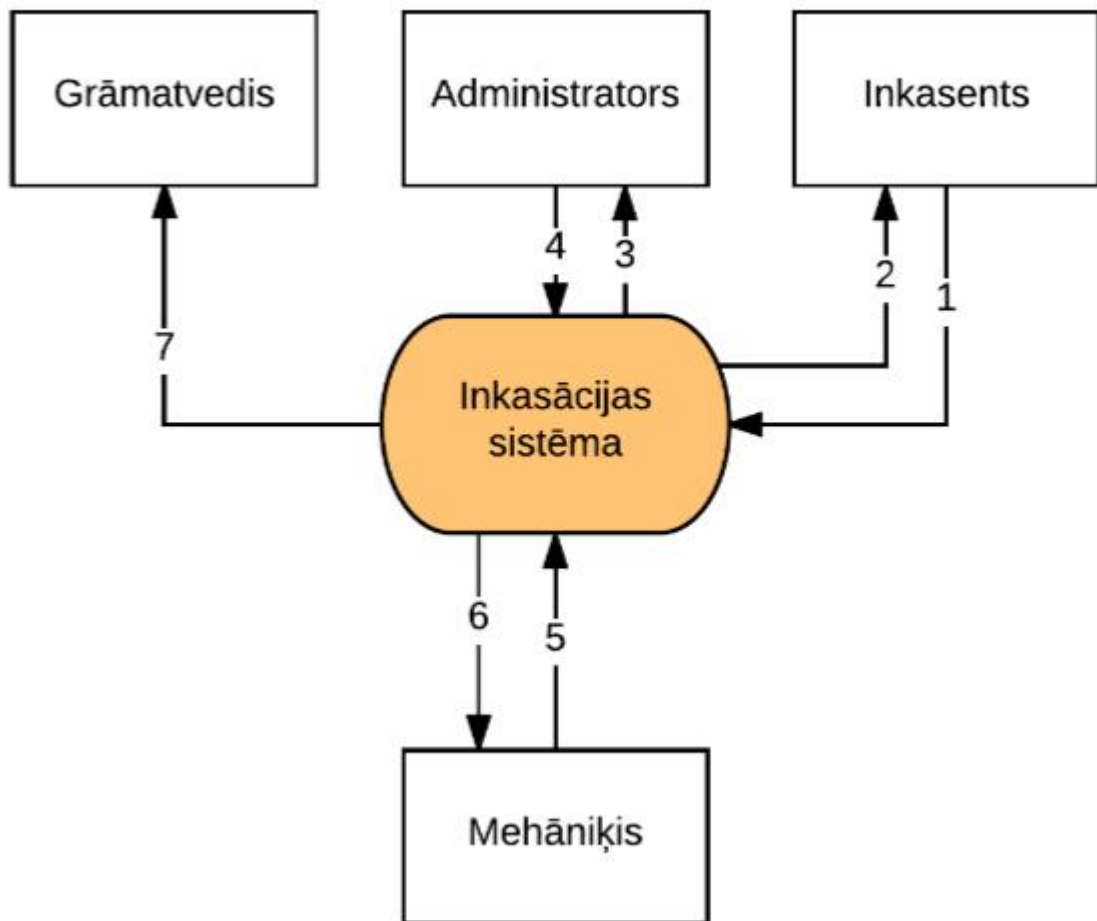
² Tie moduļi un datubāzes, kas nav iekrāsoti, nav kvalifikācijas darba ietvaros veidoti vai modificēti.

Datubāze



1. tabula, datu plūsmu diagrammu apzīmējumi

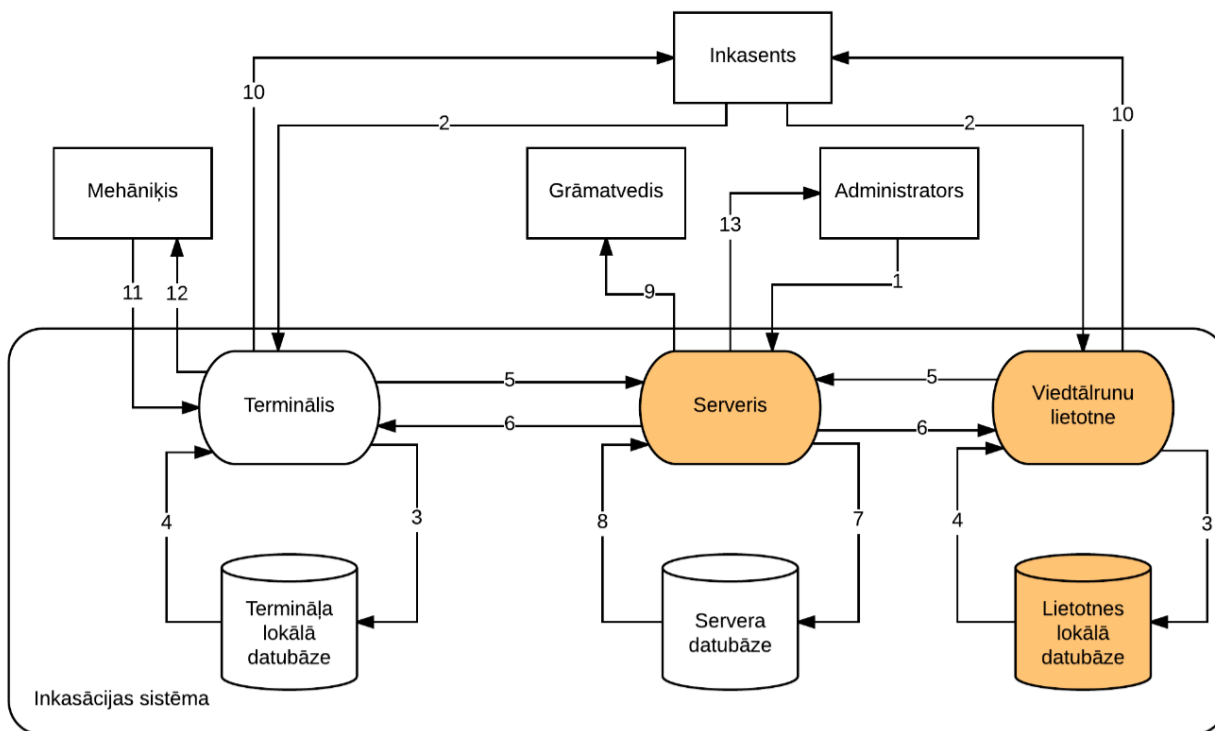
2.2.1. 0. līmenis



1. attēls, datu plūsmu diagrammas 0. līmenis

1. Inkasents veic automātu inkasāciju un ievada datus sistēmā.
2. Inkasents saņem datus par jau ievadītajiem inkasācijas aktiem.
3. Administrators saņem datus par lietotājiem un atskaitēm.
4. Sistēmas uzturētājs pievieno lietotājus, uztur sistēmas darbību, kā arī veic datu izmaiņas, ja nepieciešams.
5. Mehāniķis ievada datus par salabotajiem automātiem.
6. Mehāniķis saņem datus par jau salabotajiem automātiem.
7. Grāmatvedis saņem inkasācijas atskaites tālākai apstrādei.

2.2.2. 1. līmenis – inkasācijas sistēma

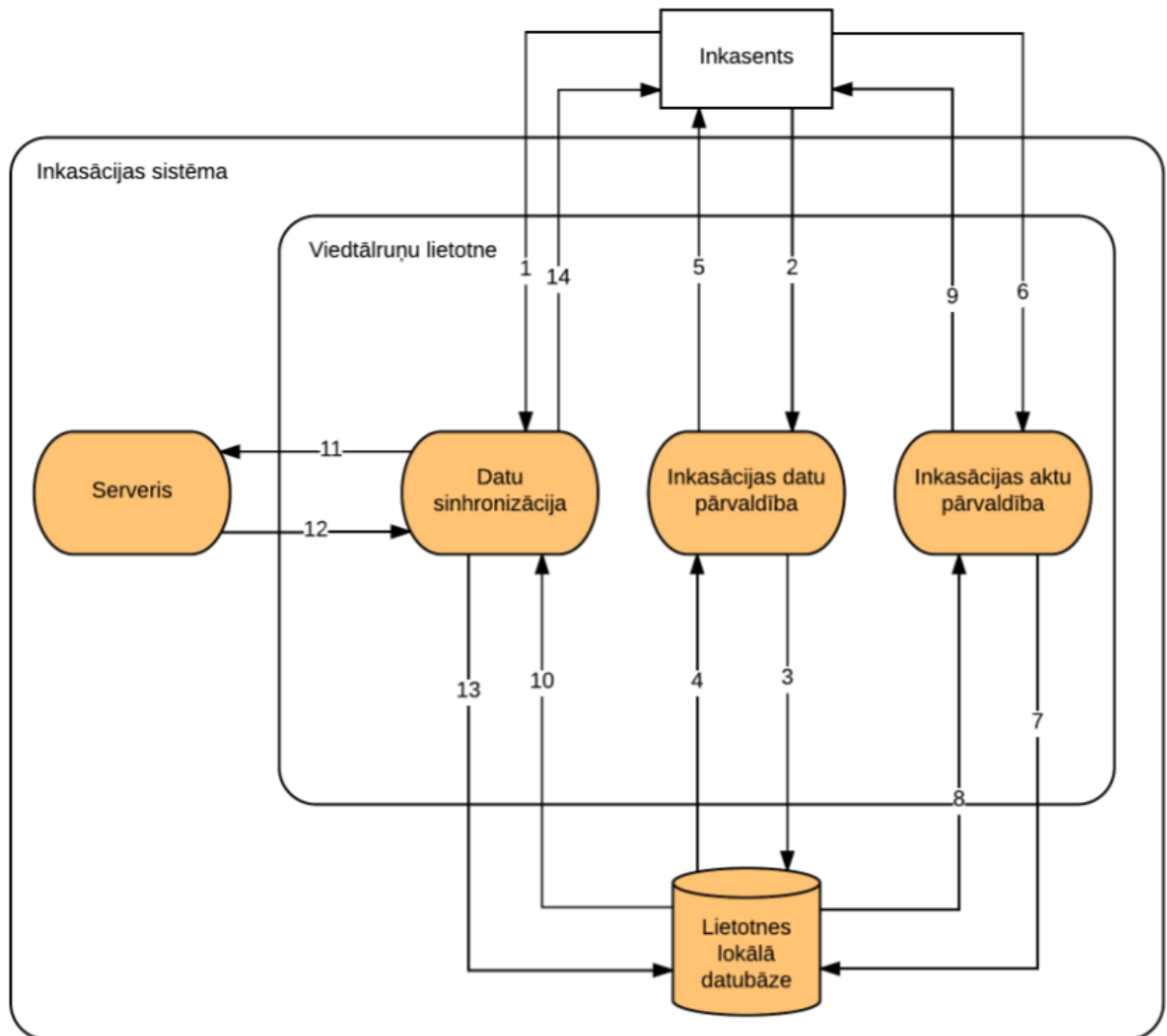


2. attēls, datu plūsmu diagrammas 1. līmenis – inkasācijas sistēma

1. Administrators pievieno lietotājus un uztur sistēmas darbību, kā arī, ja nepieciešams, veic datu izmaiņas.
2. Inkasents ievada inkasācijas datus, ko ievācis no spēļu automātiem.
3. Lietotnē/Terminālī ievadītie dati tiek pārbaudīti un, pēc pozitīvas pārbaudes, noglabāti lokālajā datubāzē.
4. Lietotnes/Termināļa datubāze atgriež pieprasītos datus.
5. Lietotne/Terminālis nosūta serverim savas lokālās datubāzes datus un veic datu pieprasījumu ar lietotāja (Lietotnes/Termināļa) autentifikācijas datiem.
6. Serveris sapako datus konkrētajam lietotājam un nosūta tos lietotnei/terminālī kā atbildi pieprasījumam.
7. Serveris noglabā datus servera datubāzē.
8. Servera datubāze atgriež pieprasītos datus.
9. Serveris nosūta grāmatvedim pieprasītos, inkasentu ievadītos, datus par konkrētu periodu.
10. Inkasents saņem datus par sevis veiktajām inkasācijām.
11. Mehāniķis ievada datus par salabotajiem automātiem.

12. Mehāniskis saņem datus par sevis salabotajiem automātiem.

2.2.3. 2. līmenis – viedtālrunu lietotne

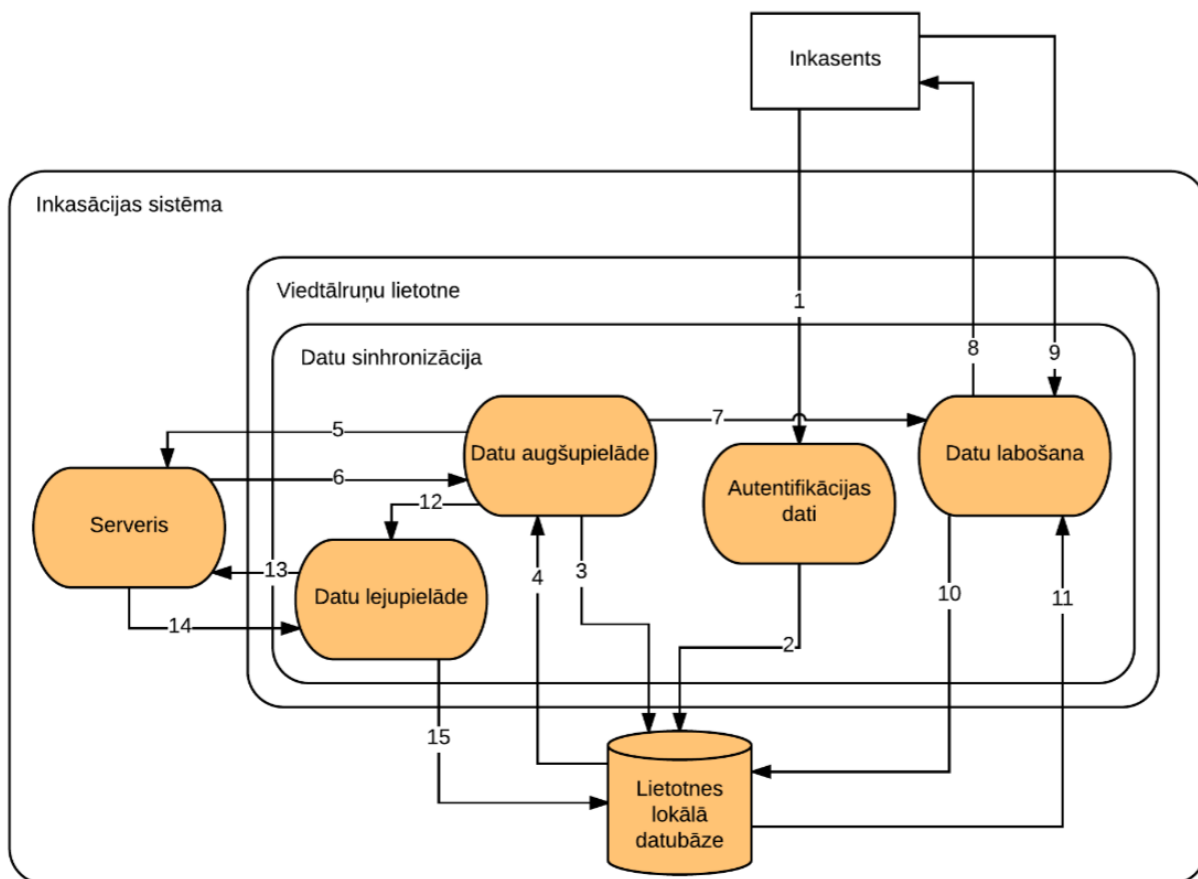


3. attēls, datu plūsmu diagrammas 2. līmenis – viedtālrunu lietotne

1. Inkasents ievada savus autentifikācijas datus vai, nepareizas datu augšupielādes gadījumā, izvēlas, kurus no ievadītajiem datiem izdzēst.
2. Inkasents ievada inkasācijas datus, ja inkasācijas dati ir pareizi formatēti, tad tie tiek sūtīti uz datubāzi.
3. Inkasācijas datu pārvaldība nosūta datus lietotnes lokālajai datubāzei, gadījumā, ja dati konkrētajā laukā nav bijuši ievadīti, tad tiek izveidots jauns ieraksts, ja dati jau ievadīti iepriekš, tad tiek atjaunots esošais ieraksts. Sīkāk skat. diagrammu **3. Līmenis – Inkasācijas datu pārvaldība**.

4. Lietotnes lokālā datubāze atgriež ievadītos inkasācijas ierakstus un adrešu sarakstus ar automātu instalācijām.
5. Inkasācijas datu pārvaldība atgriež inkasācijas objektu un instalāciju sarakstus, kā arī datus konkrētajos inkasācijas ievades laukos, ja tie jau ir bijuši aizpildīti iepriekš.
6. Inkasents ievada sev piešķirtos (neizlietos) aktu numurus.
7. Inkasācijas aktu pārvaldība nosūta datus uz lietotnes lokālo datubāzi.
8. Lietotnes lokālā datubāze atgriež visus inkasenta ievadītos inkasācijas ierakstus kopš pagājušās sinhronizācijas.
9. Inkasācijas aktu pārvaldības modulis atgriež sagatavotas atskaites konkrētai adresei uz lapām (katrā līdz 14 inkasācijas ierakstiem) drukāšanai. Sīkāk skat. diagrammu **3. Līmenis – Inkasācijas aktu pārvaldība.**
10. Lietotnes lokālā datubāze atgriež visus inkasācijas ierakstus kopš pagājušās sinhronizācijas, ja sinhronizācija nav veikta iepriekš, tad neatgriež neko.
11. Datu sinhronizācijas modulis nosūta visus, no lokālās datubāzes saņemtus, inkasācijas ierakstus kopā ar autentifikācijas datiem serverim, sīkāk skatīt diagrammu **3. līmenis – Datu sinhronizācijas apakšmodulis – Datu augšupielāde.**
12. Serveris atgriež kļūdu, ja dati nav pieņemti, bet, ja dati ir pieņemti, tad serveris atgriež jaunus, konkrētajam inkasentam paredzētos, datus.
13. Datu sinhronizācija nosūta saņemtus datus uz lietotnes lokālo datubāzi. Vecie dati tiek izdzēsti vai izmainīti, un jaunie - ievietoti datubāzē, sīkāk skatīt diagrammu **3. līmenis – Datu sinhronizācijas apakšmodulis – Datu lejupielāde.**
14. Inkasentam tiek nosūtīti dati par sinhronizācijas procesu, neveiksmīgas datu augšupielādes gadījumā inkasentam tiek atgriezts saraksts ar visiem ievadītajiem datiem.

2.2.4. 3. līmenis – datu sinhronizācija

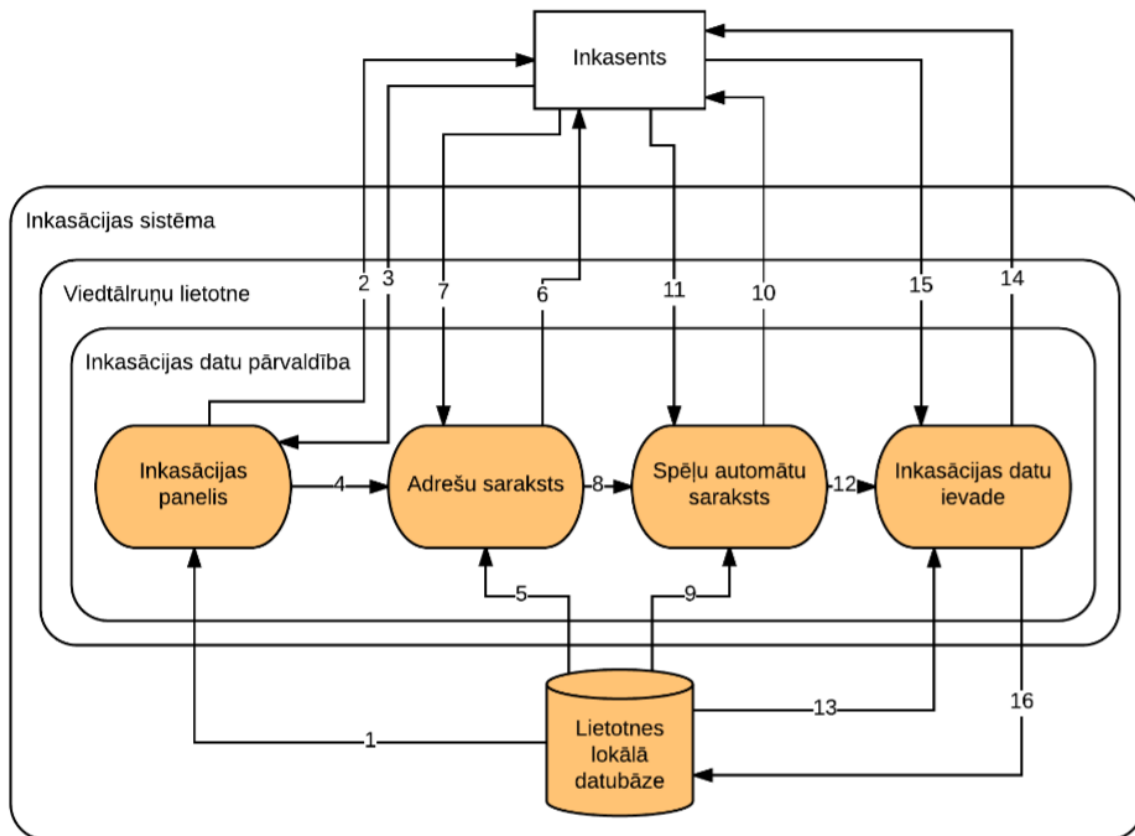


4. attēls, datu plūsmu diagrammas 3. līmenis – datu sinhronizācija

1. Inkasents ievada savus autentifikācijas datus.
2. Tiek pārbaudīts lietotājvārda formāts – tam jābūt e-pasta formā.
3. Datu augšupielādes modulis nosūta datubāzei pieprasījumu par sekojošām tabulām (tabulu definīcijas sīkāk skat. datubāzes diagrammās):
 - Encashments
 - Statements
 - StatementTemplateUsages
4. Lietotnes lokālā datubāze atgriež pieprasītos datus, kas tika ievadīti kopš pagājušās sinhronizācijas.
5. Datu augšupielādes modulis saņemtos datus pārveido serverim saprotamā formātā un nosūta tos serverim kopā ar autentifikācijas datiem.
6. Serveris pārbauda datus un, pie pozitīvas pārbaudes, noglabā tos savā datubāzē un atgriež datu augšupielādes modulim saglabāto datu identifikatorus.

7. Pie 6. punkta negatīvas pārbaudes, datu augšupielādes modulis nosūta pieprasījumu par ievadīto datu labošanu.
8. Datu labošanas modulis nosūta inkasentam visus viņa ievadītos datus.
9. Inkasents izvēlās, kurus datus no servera izdzēst.
10. Datu labošanas modulis nosūta pieprasījumu par izvēlēto datu dzēšanu.
11. Lietotnes lokāla datubāze atgriež inkasenta ievadītos datus(neskaitot tikko izdzēstos).
12. Pie 6. punkta pozitīvas pārbaudes, kā arī, ja dati nav tikuši ievadīti lietotnē iepriekš, datu augšupielādes modulis nosūta pieprasījumu par datu lejupielādi.
13. Datu lejupielādes modulis nosūta datu pieprasījumu serverim kopā ar autentifikācijas datiem.
14. Serveris atgriež visu tabulu(visas tabulas sīkāk skat. nodaļu **Datubāzes diagrammas**) datus konkrētajam lietotājam.
15. No servera saņemtie dati tiek pārveidoti lietotnes lokālajai datubāzei saprotamā formātā un nosūtīti lietotnes lokālajai datubāzei.

2.2.5. 3. līmenis – inkasācijas datu pārvaldība



5. attēls, datu plūsmu diagrammas 3. līmenis – inkasācijas datu pārvaldība

1. Lietotnes lokālā datubāze atgriež objektu skaitu un instalāciju skaitu, kas piešķirti konkrētajam lietotājam.
2. Inkasācijas panelis atgriež lietotājam viņam pieejamo automātu un inkasācijas adrešu skaitu.
3. Inkasents dod komandu pāriet uz adrešu sarakstu.
4. Inkasācijas panelis izsauc adrešu saraksta moduli.
5. Lietotnes lokālā datubāze atgriež konkrētam lietotājam pieejamās adreses.
6. Adrešu saraksta modulis atgriež inkasentam adrešu sarakstu un šādu informāciju par izvēlēto adresi:
 - uzņēmuma nosaukumu
 - uzņēmuma adresi, pasta indeksu
 - reģistrācijas numuru un datumu
 - licences numuru un datumu

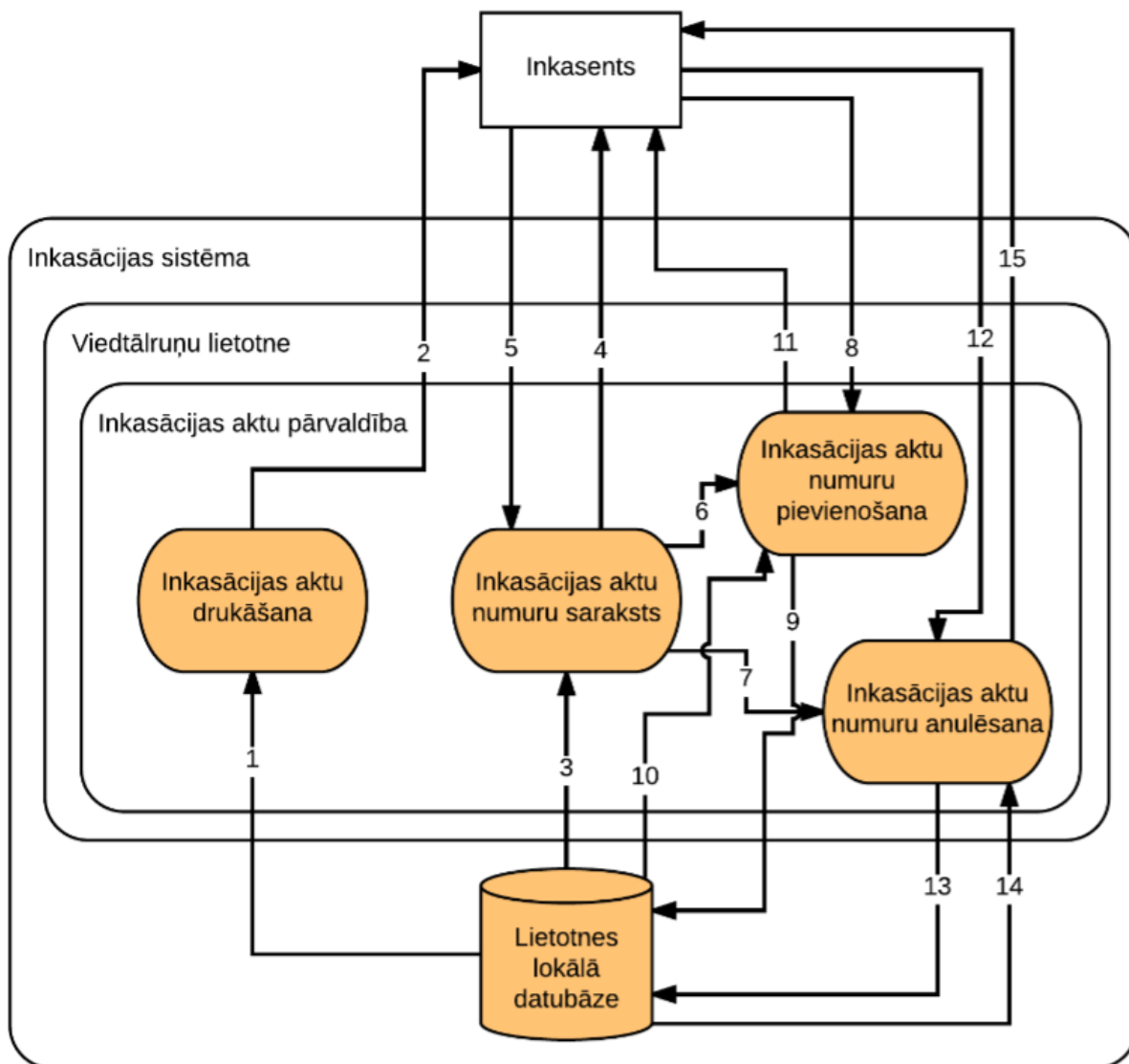
- ielas un pilsētas nosaukumus
 - pēdējo inkasācijas datumu
7. Inkasents izvēlās konkrētu adresi.
 8. Adrešu saraksta modulis nosūta spēļu automātu saraksta modulim izvēlētās adreses identifikatoru.
 9. Lietotnes lokālā datubāze atgriež informāciju par visiem spēļu automātiem konkrētajai adresei.
 10. Spēļu automātu saraksta modulis atgriež inkasentam instalāciju sarakstu un šādu informāciju par izvēlēto spēļu automāta instalāciju:
 - kods
 - programmas versija
 - programmas tips
 - ražotājs
 - likmes
 - ciparu skaitu gan mehāniskajam, gan elektroniskajam skaitītājam
 - inkasācijas summa
 11. Inkasents izvēlās konkrētu spēļu automāta instalāciju.
 12. Spēļu automātu saraksta modulis nosūta inkasācijas datu ievades modulim izvēlētās instalācijas identifikatoru.
 13. Ja inkasācija konkrētajai instalācijai ir jau notikusi kopš pagājušās sinhronizācijas, tad tiek atgriezti dati par pagājušo inkasāciju.
 14. Ja inkasācija konkrētajai instalācijai ir jau notikusi kopš pagājušās sinhronizācijas, tad inkasācijas datu ievades modulis atgriež inkasentam šādu informāciju:
 - Gan elektriskajam, gan mehāniskajam skaitītājam:
 - gan jaunie, gan iepriekšējie rādījumi:
 - iemaksātie
 - izmaksātie
 - iespēlētie kredītpunkti
 - likme
 - ieņēmums
 - viltojumi
 - uzpilde

- samazināšana
- demontāža

15. Ja inkasācija konkrētajai instalācijai ir jau notikusi kopš pagājušās sinhronizācijas, tad inkasents var izlabot inkasācijas datus, kas tiek atgriezti 14. punktā. Ja inkasācija nav notikusi, tad inkasentam dati ir jāievada no jauna (Jackpot instalācijai var ievadīt tikai izmaksātos rādījumus).

16. Inkasācijas datu ievades modulis nosūta datubāzei ievadītos vai izlabotos inkasācijas datus, ja inkasācijas ieraksts datubāzē nav, tad tiek izveidots jauns, ja ir – atjaunots esošais.

2.2.6. 3. līmenis – inkasācijas aktu pārvaldība



6. attēls, datu plūsmu diagrammas 3. līmenis – inkasācijas aktu pārvaldība

1. Lietotnes lokālā datubāze nosūta visus inkasenta ievadītos datus, kas saistīti ar konkrēto adresi, kurā inkasācija tiek veikta.
2. Inkasācijas aktu drukāšanas modulis veic aprēķinus, noformē datus, un nosūta inkasentam drukāšanas skatu ar sagatavoto informāciju.
3. Lietotnes lokālā datubāze nosūta inkasācijas aktu numuru sarakstu konkrētai adresei.
4. Inkasācijas aktu numuru saraksta modulis nosūta inkasentam šādu informāciju katram saraksta elementam:
 - sērija

- numurs
 - Statuss (izlietots vai anulēts)
5. Inkasents izvēlas pievienot, vai anulēt kādu no sev piešķirtajiem inkasācijas aktu numuriem.
 6. Ja lietotājs ir izvēlējis pievienot inkasācijas numuru, tad inkasācijas aktu numuru modulis izsauc inkasācijas aktu numuru pievienošanas moduli.
 7. Ja lietotājs ir izvēlējis anulēt inkasācijas numuru, tad inkasācijas aktu numuru modulis izsauc inkasācijas aktu numuru anulēšanas moduli.
 8. Inkasents ievada inkasācijas akta numuru un izvēlās sēriju.
 9. Ievadītie dati tiek nosūtīti lietotnes lokālajam serverim.
 10. Lietotnes lokālā datubāze atgriež datus par inkasācijas numura pievienošanu.
 11. Ja inkasācijas numurs jau ticis pievienots vai anulēts iepriekš, tad inkasentam tiek atgriezts pēdējās pievienošanas/anulēšanas datums, ja inkasentam nav iedalīts ievadītais numurs, tad tiek atgriezta kļūda, ja numurs nav pievienots/anulēts agrāk un ir iedalīts konkrētajam inkasentam, tad tiek atgriezta apstiprinoša atbilde.
 12. Inkasents ievada inkasācijas akta numuru, piezīmes un izvēlās sēriju.

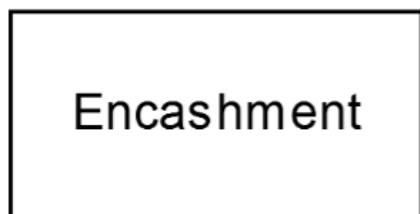
2.3. Datubāzes diagrammas

Uz šo apakšnodaļu attiecas tikai un vienīgi kvalifikācijas darba autora veidotā datubāze *React* viedtālrunu lietotnei, uz pārējiem sistēmas moduļiem (Terminālis, Serveris) šī apakšnodaļa neattiecas.

2.3.1. Konceptuālais ER modelis

Konceptuālā ER modeļa apzīmējumi:

Apzīmējums



Paskaidrojums

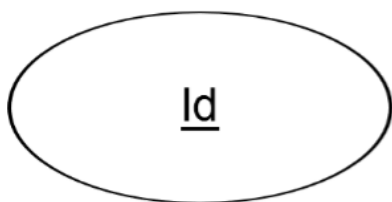
Datubāzes entītija



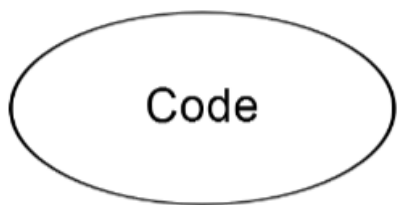
Attiecība starp datubāzes entītijām



Attiecība starp datubāzes entītijām –
daudz pret viens



Entītijas atribūts (primārā atslēga)



Entīcijas atribūts



Entīcijas atribūtu grupa

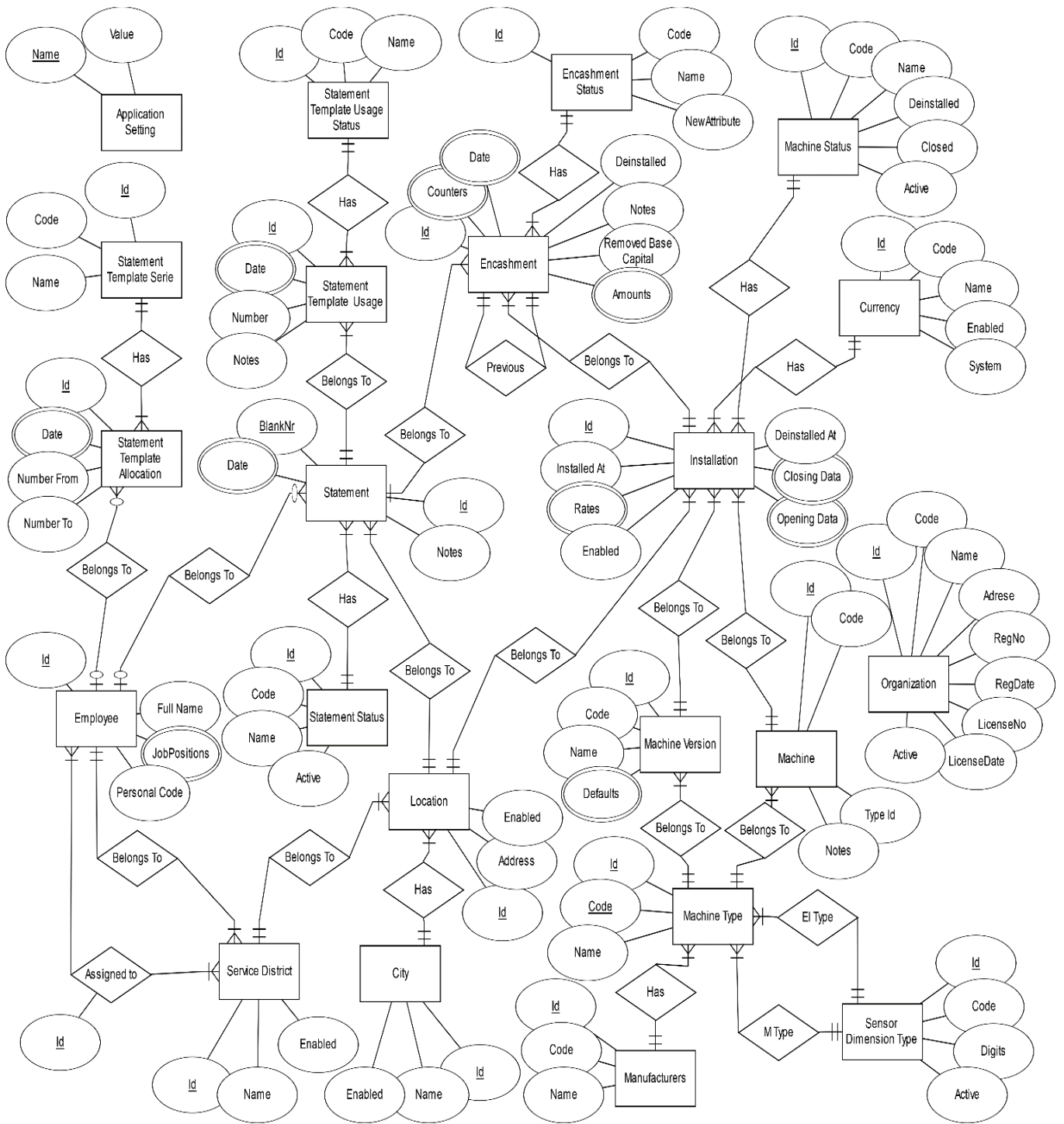
2. tabula, konceptuālā ER modeļa apzīmējumi

Lai varētu labāk pārskatīt ER modeļa diagrammu, daži no entītiju atribūtiem tika apkopoti grupās, kur katrā grupā atrodas līdzīgas nozīmes atribūti.

Atribūtu grupu atšifrējumi:

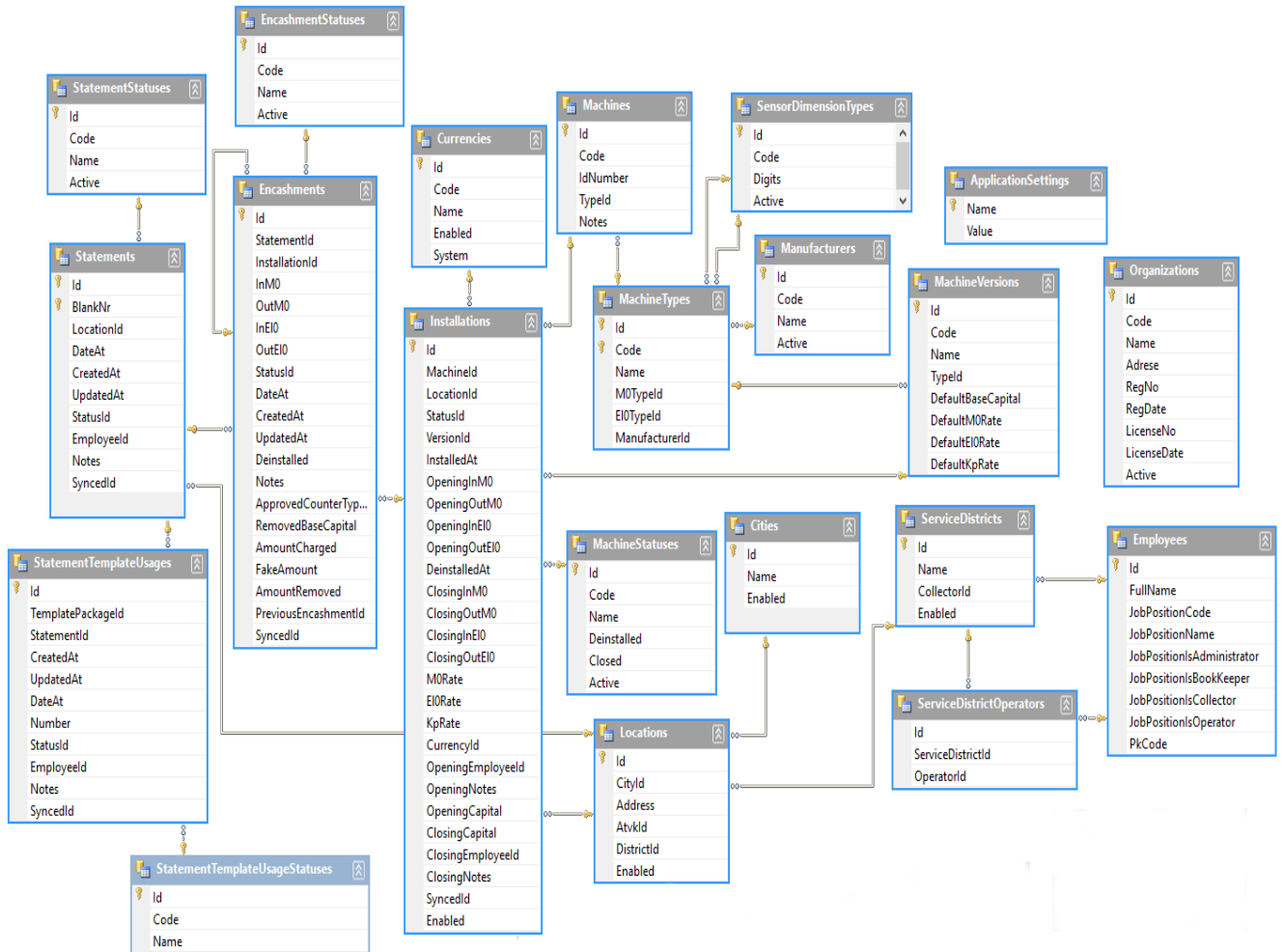
Grupa	Ietilpstošie atribūti
Date	Created at, updated at, date at.
Job Positions	Job Position: code, name, is administrator, is bookkeeper, is collector, is operator.
Counters	In mechanical, out mechanical, in electrical, out electrical.
Amounts	Amount: charged, fake, removed.
Rates	Rates: mechanical, electrical, kp.
Opening Data	Opening: in mechanical, out mechanical, in electrical out electrical, capital, notes.
Closing Data	Closing: in mechanical, out mechanical, in electrical, out electrical, capital, notes.
Defaults	Default: base capital, mechanical rate, electrical rate, kp rate.

3. tabula, konceptuālā ER modeļa entītiju atribūtu grupu atšifrējumi



7. attēls, konceptuālais entītiju relāciju modelis

2.3.2. Fiziskais ER modelis



8. attēls, fiziskais datubāzes entīciju relāciju modelis

2.3.3. Datubāzes tabulu apraksts

Datubāzes tabulas tiek aprakstītas, izmantojot šādu struktūru:

- Lauka nosaukums
- Datu tips
- NULL – vai konkrētais lauks var būt ar vērtību “null”
- Apraksts
- Piezīmes

Saīsinājumi(piezīmes):

- PK – Primary Key (primārā atslēga)
- FK – Foreign Key (ārējā atslēga)

- AI – Auto Increment (lauka vērtība automātiski pieaug par 1)

Tabula “ApplicationSettings”

Tabula satur informāciju par lietotāja autentifikācijas datiem un pēdējās sinhronizācijas laiku.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Name	varchar		Uzstādījuma identifikators	PK
Value	varchar		Uzstādījuma vērtība	

4. tabula, “ApplicationSettings” datubāzes tabulas projektējums

Tabula “Organizations”

Tabula satur informāciju par uzņēmumiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Uzņēmuma identifikators	PK
Code	varchar		Uzņēmuma kods	
Name	varchar		Uzņēmuma nosaukums	
Adrese	varchar		Uzņēmuma juridiskā adrese	
RegNo	bool		Uzņēmuma reģistrācijas numurs	
RegDate	date		Uzņēmuma reģistrācijas datums	
LicenseNo	bool		Uzņēmuma licences numurs	
LicenseDate	date		Uzņēmuma licences datums	

Active	bool		Vai uzņēmums ir aktīvs	
--------	------	--	------------------------	--

5. tabula, "Organizations" datubāzes tabulas projektējums

Tabula "StatementTemplateSeries"

Tabula satur informāciju par inkasācijas akta numura sērijām.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Sērijas identifikators	PK
Code	varchar		Sērijas kods	
Name	varchar		Sērijas nosaukums	

6. tabula, "StatementTemplateSeries" datubāzes tabulas projektējums

Tabula "StatementTemplateAllocations"

Tabula satur informāciju par inkasentam pieejamiem inkasācijas aktu numuriem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	varchar		Inkasācijas aktu numuru identifikators	PK
TemplatePackageId	varchar		Inkasācijas pakas identifikators	
SerieId	int		Sērijas identifikators	FK StatementTemplate Series (Id)
CreatedAt	timestamp		Izveidošanas laiks	
UpdatedAt	timestamp		Atjaunošanas laiks	
DateAt	date		Izveidošanas datums	
NumberFrom	int		Inkasentam piesaistītais sākuma numurs	
NumberTo	int		Inkasentam piesaistītais beigu numurs	

AssignedToId	int		Inkasenta identifikators	
--------------	-----	--	-----------------------------	--

7. tabula, "StatementTemplateAllocations" datubāzes tabulas projektējums

Tabula "StatementTemplateUsageStatuses"

Tabula satur informāciju par inkasācijas akta izlietoto numuru statusiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Statusu identifikators izlietotajiem numuriem	PK
Code	varchar		Izlietotā numura statusa kods	Iespējamās divas vērtības – IZL, ANU.
Name	varchar		Izlietotā numura statuss	Iespējamās divas vērtība – izlietots, anulēts

8. tabula, "StatementTemplateUsageStatuses" datubāzes tabulas projektējums

Tabula "StatementTemplateUsages"

Tabula satur informāciju par inkasācijas akta izlietotajiem numuriem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	varchar		Izlietoto numuru identifikators	PK, AI, Id tiek piešķirta servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinchronizācija
TemplatePackageId	varchar		Inkasācijas pakas identifikators	
StatementId	int		Inkasācijas akta identifikators	FK statements(Id)
CreatedAt	timestamp		Izveidošanas datums un laiks	

UpdatedAt	timestamp		Atjaunošanas datums un laiks	
DateAt	date		Izveidošanas datums	
Number	int		Inkasācijas akta izlietotais numurs	
StatusId	int		Inkasācijas akta izlietotā akta numura identifikators	
EmployeeId	int		Inkasenta identifikators	
Notes	text	+	Piezīmes	
SyncedId	int	+	Sinhronizācijas statuss	SyncedId tiek piešķirts servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija

9. tabula, "StatementTemplateUsages" datubāzes tabulas projektējums

Tabula "StatementStatuses"

Tabula satur informāciju par inkasācijas aktu statusiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Inkasācijas akta statusa identifikators	PK
Code	varchar		Statusa kods	
Name	varchar		Statusa nosaukums	
Active	bool		Vai statuss aktīvs	

10. tabula, "StatementStatuses" datubāzes tabulas projektējums

Tabula "Statements"

Tabula satur informāciju par inkasācijas aktiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Inkasācijas akta identifikators	PK, AI, Id tiek piešķirta servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija
BlankNr	int		Tukšs numurs	PK
LocationId	int		Inkasācijas akta izveidošanas vietas identifikators	FK Locations(Id)
CreatedAt	timestamp		Akta izveidošanas datums un laiks	
UpdatedAt	timestamp		Akta atjaunošanas datums un laiks	
DateAt	date		Akta izveidošanas datums	
StatusId	int		Inkasācijas akta statusa identifikators	FK StatementStatuses(Id)
EmployeeId	int		Inkasenta identifikators	
Notes	text	+	Piezīmes	
SyncedId	int	+	Sinhronizācijas statuss	SyncedId tiek piešķirta servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija

11. tabula, "StatementTemplateUsages" datubāzes tabulas projektējums

Tabula "EncashmentStatuses"

Tabula satur informāciju par inkasācijas statusiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Inkasācijas statusa identifikators	PK
Code	varchar		Statusa kods	
Name	varchar		Statusa nosaukums	
Active	bool		Vai statuss aktīvs	

12. tabula, "EncashmentStatuses" datubāzes tabulas projektējums

Tabula "Encashments"

Tabula satur informāciju par inkasācijām.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Inkasācijas identifikators	PK, AI, Id tiek piešķirta servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija
StatementId	int		Inkasācijas akta identifikators	FK Statements(Id)
InstallationId	int		Spēļu automāta instalācijas identifikators	FK Installations(Id)
InM0	int		"In" mehāniska skaitītāja vērtība	
OutM0	int		"Out" mehāniskā skaitītāja vērtība	
InE10	int		"In" elektroniskā skaitītāja vērtība	
OutE10	int		"Out" elektroniskā skaitītāja vērtība	

StatusId	int		Inkasācijas statusa identifikators	FK EncashmentStatuses(Id)
CreatedAt	timestamp		Inkasācijas izveidošanas datums un laiks	
UpdatedAt	timestamp		Inkasācijas atjaunošanas datums un laiks	
DateAt	date		Inkasācijas izveidošanas datums	
Notes	text	+	Piezīmes	
ApprovedCounterTypeId	int		Skaitītāja tips	Ir iespējamas tikai divas vērtības: mehāniskais – 1, elektroniskais – 2.
RemovedBaseCapital	int		Izņemtais sākuma kapitāls	
AmountCharged	int		Inkasācijas laikā ieliktais naudas daudzums	
FakeAmount	int		Inkasācijas laikā atrastie naudas viltojumi	
AmountRemoved	int		Inkasācijas laikā ieliktais naudas daudzums	
PreviousEncashmentId	int		Iepriekšējās inkasācijas identifikators	FK Encashments(Id)
SyncedId	int	+	Sinhronizācijas statuss	SyncedId tiek piešķirta servera datubāzes

				ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija
--	--	--	--	---

13. tabula, "Encashments" datubāzes tabulas projektējums

Tabula "MachineStatuses"

Tabula satur informāciju par automātu instalāciju statusiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Inkasācijas identifikators	PK
Code	varchar		Statusa kods	
Name	varchar		Statusa nosaukums	
Deinstalled	bool		Vai automāts ir atinstalēts	
Closed	bool		Vai automāts ir aizvērts	
Active	bool		Vai automāts ir aktīvs	

14. tabula, "MachineStatuses" datubāzes tabulas projektējums

Tabula "Currencies"

Tabula satur informāciju par automātu instalācijai piešķirto valūtu.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Valūtas identifikators	PK
Code	varchar		Valūtas kods	
Name	varchar		Valūtas nosaukums	
Enabled	bool		Vai valūta ir aktīva	

System	bool		Vai tā ir noklusētā valūta	
--------	------	--	----------------------------	--

15. tabula, "Currencies" datubāzes tabulas projektējums

Tabula "Installations"

Tabula satur informāciju par spēļu automātu instalācijām.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Instalācijas identifikators	PK, Id tiek piešķirta servera datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija
MachineId	int		Automāta identifikators	FK Machines(Id)
LocationId	int		Instalācijas vietas identifikators	FK Locations(Id)
StatusId	int		Instalācijas statusa identifikators	FK MachineStatuses(Id)
VersionId	int		Instalācijas automāta versijas identifikators	FK MachineVersions(Id)
InstalledAt	timestamp		Instalācijas uzstādīšanas datums un laiks	
OpeningInM0	int		Automāta instalācijas "In" mehāniskā skaitītāja vērtība uzstādīšanas brīdī	
OpeningOutM0	int		Automāta instalācijas "Out"	

			mehāniskā skaitītāja vērtība uzstādīšanas brīdī	
OpeningInEI0	int		Automāta instalācijas “In” elektroniskā skaitītāja vērtība uzstādīšanas brīdī	
OpeningOutEI0	int		Automāta instalācijas “In” elektroniskā skaitītāja vērtība uzstādīšanas brīdī	
DeinstalledAt	date		Instalācijas noņemšanas datums un laiks	
ClosingInM0	text	+	Automāta instalācijas “In” mehāniskā skaitītāja vērtība demontāžas brīdī	
ClosingOutM0	int		Automāta instalācijas “Out” mehāniskā skaitītāja vērtība demontāžas brīdī	
ClosingInEI0	int		Automāta instalācijas “In” elektroniskā skaitītāja vērtība demontāžas brīdī	

ClosingOutEI0	int		Automāta instalācijas “In” elektroniskā skaitītāja vērtība demontāžas brīdī	
MORate	float		Instalācijas mehāniska skaitītāja likme	
EI0Rate	float		Instalācijas elektroniskā skaitītāja likme	
KpRate	float		Kreditpunktu nomināls	
CurrencyId	int	+	Valūtas identifikators	FK Currencies(Id)
OpeningEmployeeId	int		Darbinieka identifikators, kurš ir atvēris instalāciju	
OpeningNotes	text		Instalācijas atvēršanas piezīmes	
OpeningCapital	int		Instalācijas atvēršanas kapitāls	
ClosingEmployeeId	int		Darbinieka identifikators, kurš ir aizvēris instalāciju	
ClosingNotes	text		Instalācijas aizvēršanas piezīmes	
ClosingCapital	int		Instalācijas aizvēršanas kapitāls	
SyncedId	int		Sinhronizācijas identifikators	SyncedId tiek piešķirta servera

				datubāzes ieraksta Id vērtība, kad veiksmīgi beigusies sinhronizācija
Enabled	bool		Vai automāta instalācija ir pieejama pašreizējam darbiniekam	

16. tabula, "Installations" datubāzes tabulas projektējums

Tabula "Machines"

Tabula satur informāciju par spēļu automātiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Automāta identifikators	PK
Code	varchar		Automāta kods	
IdNumber	int		Automāta identifikācijas numurs	
TypeId	int		Automāta tipa identifikators	FK MachineTypes(Id)
Notes	text		Piezīmes	

17. tabula, "Machines" datubāzes tabulas projektējums

Tabula "MachineTypes"

Tabula satur informāciju par spēļu automātu tipiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Tipa identifikators	PK
Code	varchar		Tipa kods	PK
Name	varchar		Tipa nosaukums	
M0TypeId	int		Automāta mehāniskā skaitītāja identifikators	FK SensorDimensionTypes(Id)
EI0TypeId	int		Automāta elektroniskā skaitītāja identifikators	FK SensorDimensionTypes(Id)
ManufacturerId	int		Automāta ražotāja identifikators	FK Manufacturers(Id)

18. tabula, "MachineTypes" datubāzes tabulas projektējums

Tabula "SensorDimensionTypes"

Tabula satur informāciju par spēļu automātu skaitītāju ciparu skaitu.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Skaitītāja identifikators	PK
Code	varchar		Skaitītāja kods	
Digits	int		Automāta skaitītāja ciparu skaits	

Active	bool		Vai konkrētais skaitītāju tips ir aktīvs	
--------	------	--	--	--

19. tabula, "SensorDimensionTypes" datubāzes tabulas projektējums

Tabula "Manufacturers"

Tabula satur informāciju par spēļu automātu ražotāju.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Ražotāja identifikators	PK
Code	varchar		Ražotāja kods	
Name	varchar		Ražotāja nosaukums	
Active	bool		Vai ražotājs ir aktīvs	

20. tabula, "Manufacturers" datubāzes tabulas projektējums

Tabula "MachineVersions"

Tabula satur informāciju par spēļu automātu programmatūras versijām.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Versijas identifikators	PK
Code	varchar		Versijas kods	
Name	varchar		Versijas nosaukums	
TypeId	int		Spēļu automāta tipa identifikators	FK MachineTypes(Id)
DefaultBaseCapital	int		Noklusētas sākuma kapitāls	
DefaultMORate	float		Noklusētā mehānisko skaitītāju likme	

DefaultElORate	float		Noklusētā elektronisko skaitītāju likme	
DefaultKpRate	float		Noklusētais Kredītpunkta nomināls	

21. tabula, "MachineVersions" datubāzes tabulas projektējums

Tabula "Cities"

Tabula satur informāciju par pilsētām, kur atrodas spēļu automātu instalācijas.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Pilsētas identifikators	PK
Name	varchar		Pilsētas nosaukums	
Enabled	int		Vai pilsētā atrodas instalācijas, kuras jāapkalpo pašreizējam darbiniekam	

22. tabula, "Cities" datubāzes tabulas projektējums

Tabula "Locations"

Tabula satur informāciju par pilsētām, kur atrodas spēļu automātu instalācijas.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Adreses identifikators	PK
CityId	int		Pilsētas identifikators	
Address	int		Adrese	

AtvkId	int		Administratīvo teritoriju un teritoriālo vienību klasifikators	
DistrictId	int		Rajona identifikators	
Enabled	bool		Vai adresē atrodas instalācijas, kuras jāapkalpo pašreizējam darbiniekam	

23. tabula, "Locations" datubāzes tabulas projektējums

Tabula "ServiceDistricts"

Tabula satur informāciju par rajoniem, kuri jāapkalpo inkasentam.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Adreses identifikators	PK
Name	int		Rajona nosaukums	
CollectorId	int		Inkasenta identifikators	
Enabled	int		Vai rajonā atrodas instalācijas, kuras jāapkalpo pašreizējam darbiniekam	

24. tabula, "ServiceDistricts" datubāzes tabulas projektējums

Tabula "ServiceDistrictOperators"

Tabula satur informāciju par to, kādi darbinieki ir piesaistīti kādiem rajoniem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Adrese identifikators	PK
ServiceDistrictId	int		Rajona identifikators	
OperatorId	int		Inkasenta identifikators	

25. tabula, "ServiceDistrictOperators" datubāzes tabulas projektējums

Tabula "Employees"

Tabula satur informāciju par darbiniekiem.

Lauka nosaukums	Datu tips	NULL	Apraksts	Piezīmes
Id	int		Darbinieka identifikators	PK
FullName	varchar		Darbinieka vārds un uzvārds	
JobPositionCode	varchar		Darbinieka amata kods	
JobPositionName	varchar		Darbinieka amata nosaukums	
JobPositionIsAdministrator	bool		Vai darbinieks ir administrators	
JobPositionIsBookKeeper	bool		Vai darbinieks ir grāmatvedis	
JobPositionIsCollector	bool		Vai darbinieks ir inkasents	
JobPositionIsOperator	bool		Vai darbinieks ir automātu operators(labotājs)	

PkCode	varchar		Darbinieka Personas Kods	
--------	---------	--	-----------------------------	--

26. tabula, "Employees" datubāzes tabulas projektējums

2.4. Moduļu projektējums

Visi zemāk minētie moduļi tika veidoti izmantojot *ReactJS* ietvaru, kas atrodas *React-Native* interneta pārlūkprogrammas skatā un kas, savukārt, ir ietverts viedtālrunu lietotnē. Interneta pārlūkprogrammas skats nodrošina lietotni ar *IndexedDB* datubāzi, ko kontrolē *Lovefield* ietvars.

2.4.1. Datu sinhronizācija

Lietotnes datu sinhronizācijas modulis ir paredzēts datu apmaiņai ar inkasācijas sistēmas serveri. Datu apmaiņai tiek izmantots *HTTPS POST* pieprasījums, kas tiek nosūtīts uz inkasācijas sistēmas servera *URL*, ko nodrošina *Axios HTTP* klients. Lai datu sinhronizāciju nevarētu veikt nesankcionēta persona, datu pieprasījums tiek nosūtīts kopā ar inkasenta autentifikācijas datiem.

Datu sinhronizācijas modulis tiek loģiski iedalīts 2 galvenajos apakšmoduļos – datu lejupielāde un datu augšupielāde. Katrs no šiem apakšmoduļiem nosūta *HTTPS POST* pieprasījumu serverim ar inkasenta autentifikācijas datiem, taču pieprasījuma ķermenī tiek ietverta atšķirīga informācija. Moduļu darbības secība ir sekojoša - datu augšupielāde, ja inkasents ir ievadījis datus, pēc tam - datu lejupielāde. Datu labošanas apakšmodulis ļauj lietotājam izdzēst konfliktējošos inkasācijas datus, bet autentifikācijas datu apakšmodulis - noglabāt autentifikācijas datus uz lokālās datubāzes. Datu sinhronizācijas modulis atbild par datu pārbaudi un rediģēšanu sekojošajos skatos:

- Datu labošana (ievade)
- Autentifikācija (ievade)
- Sinhronizēšana (izvade)

Datu augšupielādes apakšmodulis

Šis apakšmodulis ir paredzēts datu nosūtīšanai uz serveri. Lai sekmīgi nosūtītu datus, *HTTPS POST* pieprasījuma ķermenī ir jābūt speciāli formatētiem datiem, kas saņemti no viedtālrunu lietotnes lokālās datubāzes (funkciju, kas sagatavo datus nosūtīšanai sīkāk skat. **Pielikums - XML faila sagatavošana datu nosūtīšanai**). Dati, kas tiek nosūtīti uz serveri ir visi, ko inkasents ir ievadījis lietotnē kopš pagājušās sinhronizācijas. Nosūtīšanai uz serveri tiek sagatavoti dati no sekojošām tabulām:

- Encashments
- Statements

- StatementUsages

Visas augstāk minēto tabulu rindas, kuru laukos syncedId vērtība ir ‘null’ tiek sagatavotas nosūtīšanai uz serveri. Serverim atbildot pozitīvi, tiek saņemti iesūtīto datu identifikatori un serverī saglabāto datu identifikatori. Datu augšupielādes apakšmodulim apstrādājot servera atbildi, lokālie identifikatori un, visi syncedId lauki, kam vērtība ir ‘null’, tiek nomainīti uz servera saņemtajiem identifikatoriem, tādējādi atzīmējot lokālajā datubāzē, ka šie dati jau ir tikuši nosūtīti. Pie servera negatīvas atbildes, lietotājam tiek dotas tiesības izdzēst nepareizi ievadītos inkasācijas datus.

Datu lejupielādes apakšmodulis

Šis apakšmodulis ir paredzēts datu saņemšanai no servera. Lai sekmīgi saņemtu datus, *HTTPS POST* pieprasījuma ķermenī ir jābūt ietvertam viedtālrunu lietotnes lokālajam laikam. Pieprasījuma atbildē serveris nosūta datus, kas paredzēti visām **Fiziskajā ER modelī** uzskaitītajām tabulām, izņemot ApplicationSettings – tā ir domāta autentifikācijas datu un pēdējā sinhronizācijas laika uzglabāšanai. Saņemtais datu formāts, šī apakšmoduļa ietvaros, tiek pārveidots par *Lovefield* sprotamu objektu kopumu, kas ir viegli ievietojams datubāzē.

2.4.2. Inkasācijas datu pārvaldība

Lietotnes inkasācijas datu pārvaldības modulis ir paredzēts inkasācijas datu ievadei, labojumiem un pārbaudei, kā arī navigācijai starp automātu adresēm un instalācijām. Visus inkasācijas datus ir iespējams izlabot un izdzēst, vēlreiz apskatot ievadītos datus konkrētajai instalācijai. Inkasācijas datu pārvaldība ir pieejama tikai pēc datu sinhronizācijas.

Lai ievadītu inkasācijas datus ir jāizdara adreses izvēle un tad instalācijas izvēle konkrētajai adresei. Instalācijas izvēli var arī veikt, izmantojot svītrkodu skanētāju, nolasot svītrkodu, kas atrodas uz spēļu automāta. Šis modulis gādā par datu nosūtīšanu (datu atlasī) un datu ievades pārbaudi sekojošajos skatos:

- Inkasācijas paneļa skats (izvade)
- Adrešu skats (izvēle)
- Instalāciju skats (izvēle)
- Inkasācijas dati (ievade)
- Naudas uzpilde (ievade)

Visos izvēles skatos attēlojamā informācija ir dinamiska – saņemta no lietotnes lokālās datubāzes. Ievades skatos tiek attēlota, konkrētai instalācijai iepriekš ievadītā, neizdzēstā informācija kopš pagājušās sinhronizācijas.

2.4.3. Inkasācijas aktu pārvaldība

Lietotnes inkasācijas aktu pārvaldības modulis ir paredzēts inkasācijas aktu drukāšanai, numuru ievadei un anulēšanai. Inkasācijas aktu pārvaldība ir pieejama tikai pēc datu sinhronizācijas. Lai ievadītu vai anulētu inkasentam piešķirto inkasācijas numuru, inkasentam ir nepieciešams ievadīt inkasācijas datus vismaz vienai instalācijai konkrētajai adresei, kurai inkasents vēlas pievienot inkasācijas akta numuru.

Inkasācijas akti tiek sagatavoti katrai adresei atsevišķi. Drukājot, katrā lapā tiek attēloti maksimums 14 inkasācijas ierakstu un kopējie rādījumi, “Jackpot” inkasācijas netiek attēlotas kā ieraksti, taču tiek ieskaitītas, kad tiek rēķināti kopējie rādījumi. Uz katras izdrukātās lapas inkasents uzspiež zīmogu, kur, ar katru zīmoga spiedienu, mainās inkasācijas aktu numurs. Katrai lapai lietotnē ir jāievada uz lapas uzspiestais inkasācijas aktu numurs. Lapas formāta paraugu var redzēt pielikumā - **Inkasācijas akta izdrukas paraugs**.

Inkasācijas aktu pārvaldības modulis rūpējas par datu atlasīšanu un rediģēšanu, kā arī ievades pārbaudi sekojošiem skatiem:

- IA numuru saraksts (izvade)
- IA numuru pievienošana (ievade)
- IA numuru anulēšana (ievade)

2.5. Lietotāja ekrāna formu projektējums

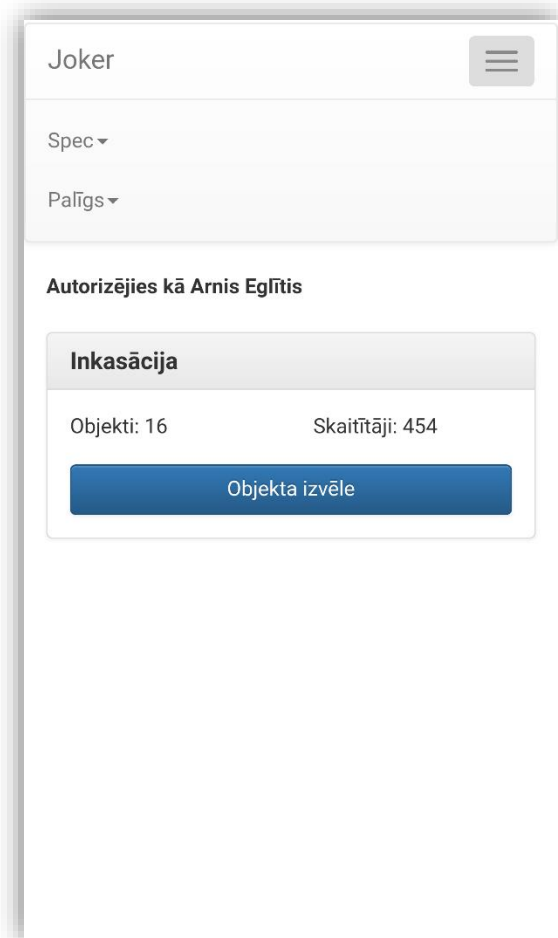
Ekrāna formas ir nepieciešamas visiem izstrādātajiem moduļiem un tās ir paredzētas “Joker” spēļu zāļu inkasentiem, kuri lieto inkasācijas sistēmas viedtālrunu lietotni.

Tā kā ekrāna formas attiecas tikai un vienīgi uz viedtālrunu lietotni, tad arī visas šīs formas tiek veidotas pēc līdzīga dizaina.

Visiem skatiem ir pieejama viena un tā pati rīkjoslā. Rīkjoslā ir pieejamas saites uz 3 skatiem: **Par Lietotni, Datu Sinhronizācija, Autentifikācijas dati**. Pārējie skati ir pieejami, izmantojot saites konkrētajā skatā.

2.5.1. Inkasācijas panelis

Uzskatāmības dēļ šajā skatā ir redzama rīkjosla atvērtā veidā. Zem sadaļas “Spec” tā satur saites uz šādiem skatiem: **Datu sinhronizācija**, **Autentifikācijas dati**, taču zem sadaļas “Palīgs” – **Par lietotni**.

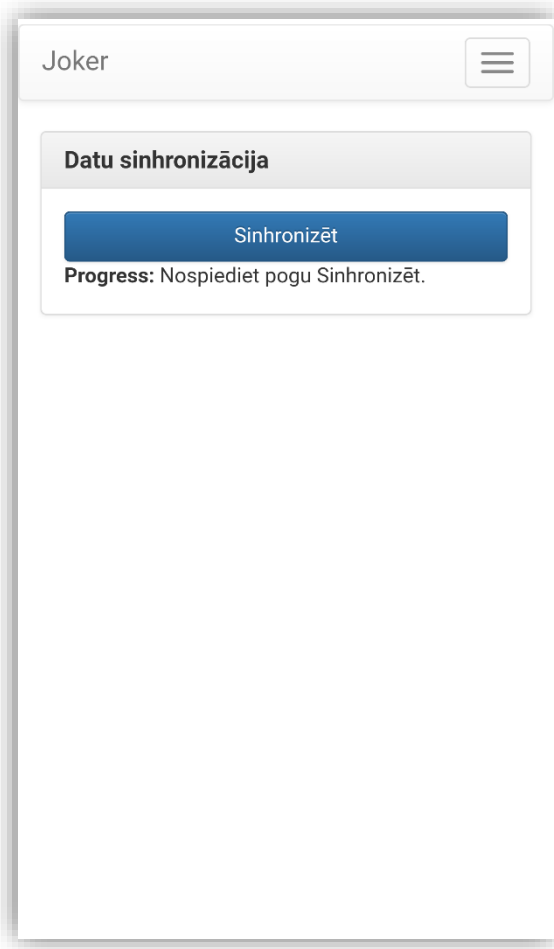


9. attēls, Inkasācijas paneļa skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Inkasenta vārds un uzvārds
- Inkasentam piešķirto adresu un instalāciju skaitu
- Poga “Objekta izvēle”, kas ir saite uz **Adreses izvēle** skatu

2.5.2. Datu sinhronizācija

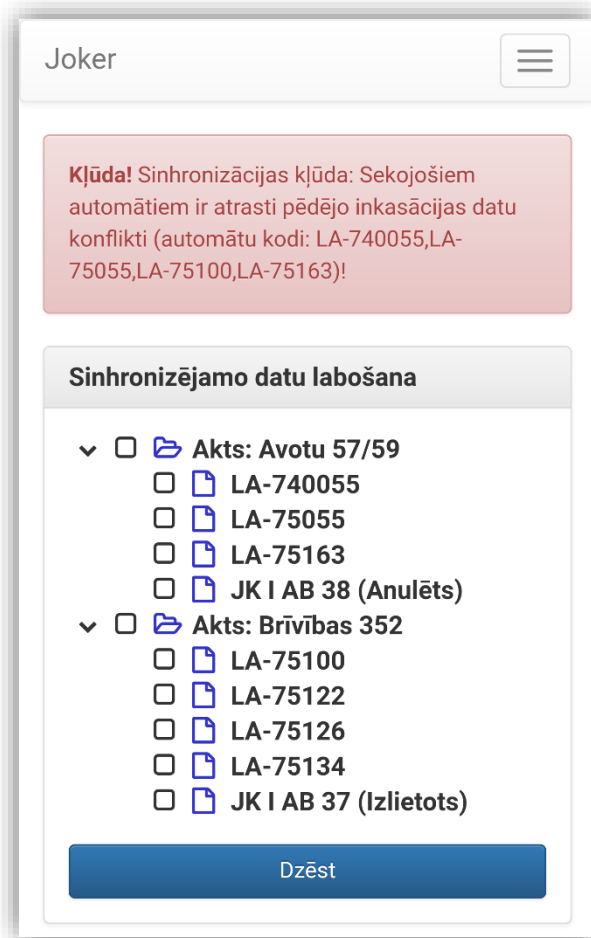


10. attēls, Datu sinhronizācijas skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Progress, kas mainās atkarībā no sinhronizācijas stāvokļa
- Poga "Sinhronizēt", kas sāk datu sinhronizāciju
- Poga "Rediģēt sinhr. datus" ar attiecīgo kļūdas paziņojumu, ja sinhronizācijas dati nav pareizi ievadīti, kas ir saite uz **Sinhronizējamo datu labošana** skatu

2.5.3. Sinhronizējamo datu labošana

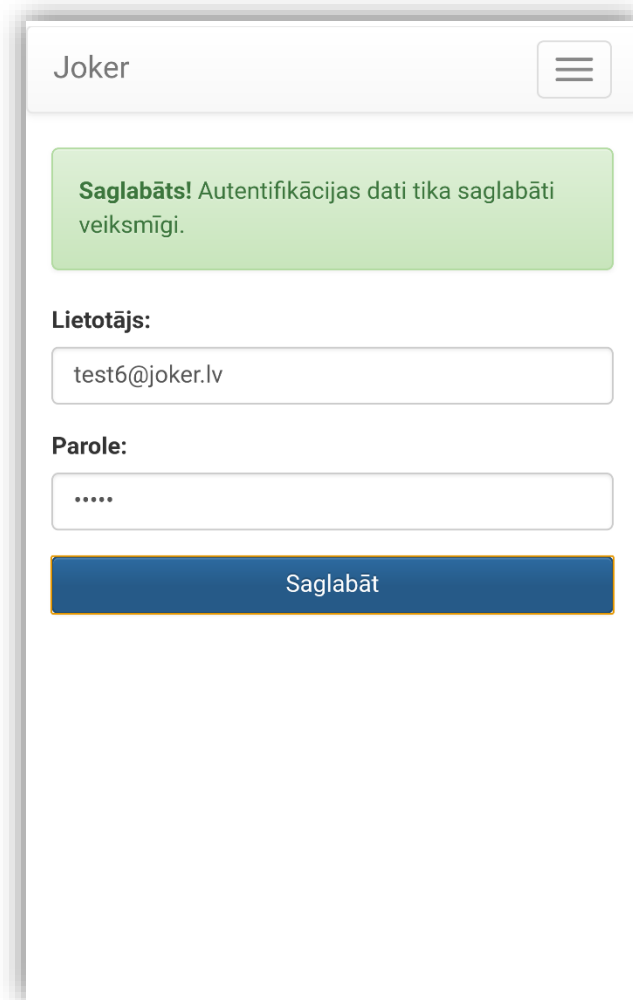


11. attēls, Sinhronizējamo datu labošanas skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Kļūdas paziņojums, kas apraksta nepareizi ievadītos datus
- Saraksts ar visiem ievadītajiem datiem, kurus iespējams izdzēst
- Poga "Dzēst", kas ļauj izdzēst atzīmētos datus

2.5.4. Autentifikācijas dati



The screenshot shows a mobile application interface for 'Joker'. At the top, the name 'Joker' is displayed next to a hamburger menu icon. Below this, a green message box contains the text: 'Saglabāts! Autentifikācijas dati tika saglabāti veiksmīgi.' (Saved! Authentication data was saved successfully.) Underneath the message, there are two input fields: 'Lietotājs:' (User) containing 'test6@joker.lv' and 'Parole:' (Password) containing '.....'. At the bottom of the form is a blue button labeled 'Saglabāt' (Save).

12. attēls, Sinhronizējamo datu labošanas skats

Šajā skatā papildus rīkjoslai tiek attēlota:

- Datu ievadforma:
 - Lietotājvārds
 - Parole
- Poga “Saglabāt”, kas saglabā datus
- Paziņojums par datu saglabāšanas stāvokli, kas parādās pēc datu saglabāšanas

2.5.5. Adrešes izvēle

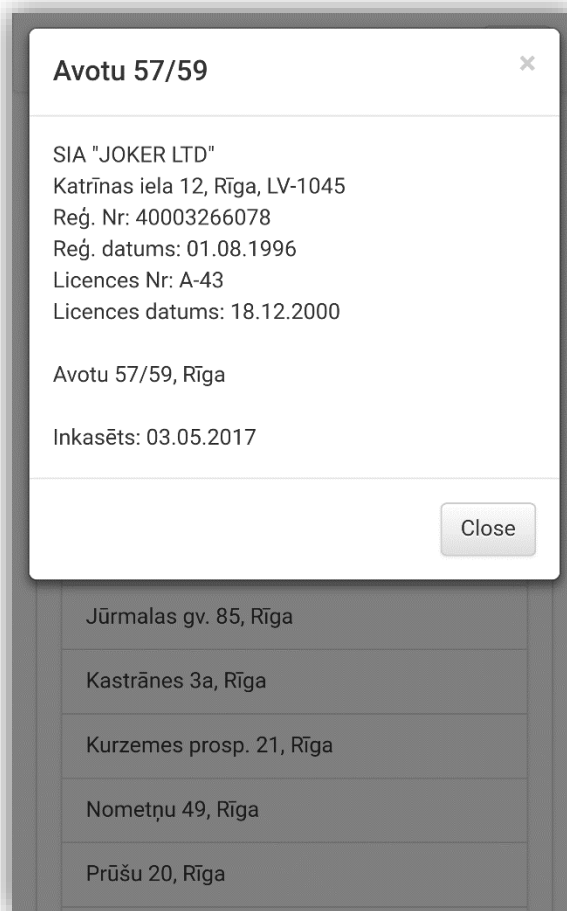


14. attēls, Adrešes izvēles skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Adrešu saraksts ar saitēm uz katras adrešes instalācijām:
 - Poga “Izvēlēties”, kas ir saite uz konkrētās adrešes **Instalāciju izvēle** skatu
 - Poga “Info”, kas parāda izvēlētās adrešes informāciju – **Adrešes informācija** skats

2.5.6. Adrešes informācija

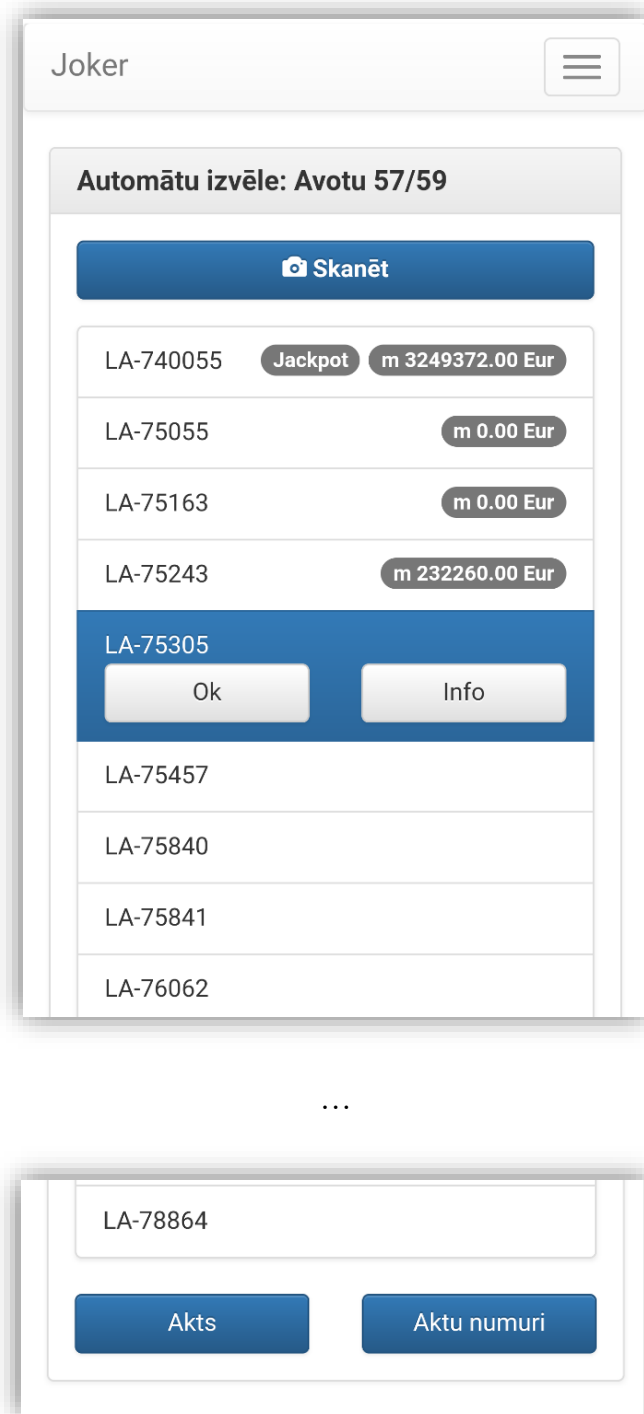


15. attēls, Adrešes informācijas skats

Šajā skatā tiek attēlots:

- Uzņēmuma nosaukums
- Uzņēmuma juridiskā adrese
- Reģistrācijas dati
- Licences dati
- Adrese
- Pēdējās inkasācijas datums norādītajā adresē
- Poga "Close", kas aizver skatu

2.5.7. Instalāciju izvēle



16. attēls, Instalāciju izvēles skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Instalāciju adrese
- Instalāciju saraksts ar saitēm uz inkasācijas ievades formu:

- Poga “Izvēlēties”, kas ir saite uz konkrētās instalācijas **Skaitītāju dati** skatu
- Poga “Info”, kas parāda izvēlētajā instalācijas informāciju – **Instalācijas informācija** skatu
- Ievadīto inkasāciju ieņēmumi
- Poga “Skanēt”, kas ļauj noskanēt svītrkodu
- Poga “Akts”, kas ļauj izdrukāt inkasācijas aktu (pielikums – **Inkasācijas akta izdrukas paraugs**)
- Poga “Aktu numuri”, kas ir saite uz **IA numuru saraksts** skatu konkrētai adresei

2.5.8. Instalācijas informācija



17. attēls, Instalācijas informācijas skats

Šajā skatā tiek attēlots:

- Automāta nosaukums
- Automāta programmatūras nosaukums
- Automāta programmatūras tips
- Automāta ražotāja nosaukums
- Likmes, kas ietver:
 - Elektroniskā skaitītāja likmi
 - Mehāniska skaitītāja likmi
 - Spēļu automāta dalības maksu
- Zīmes, kas parāda automāta skaitītāju ciparu skaitu
- Summa, kas parāda inkasācijas ieņēmumus

- Poga “Close”, kas aizver skatu

2.5.9. Inkasācijas datu ievade

Inkasācijas ievade tiek sadalīta divos atsevišķos skatos – **Skaitītāju dati**, kurā tiek ievadīta informācija par konkrētās instalācijas skaitītāju stāvokli inkasācijas brīdī, un **Naudas uzpilde**, kurā tiek ievadīta informācija par automātā esošajiem viltojumiem, kā arī izņemto un uzpildīto naudas daudzumu.

2.5.9.1. Skaitītāju dati

Joker ☰

LA-75243

	IN	OUT
Meh.	12	123
Elek.	21	321

Ok Ei Labot Tālāk

	Meh. IN	Meh. OUT
Jaunie rād.	0000012	0000123
Iepriekš rād.	6535585	6767956
Iespēlētie kp.	232260	
Likme	1	
Ieņēmums	232260.00	Eur

...

	EI. IN	EI. OUT
Jaunie rād.	000000021	000000321
Iepriekš rād.	568986868	989868689
Iespēlētie kp.	420881521	
Likme	0.01	
Ieņēmums	4208815.21	Eur

18. attēls, Skaitītāju datu ievades skats

Šajā skatā papildus rīkjoslai tiek attēloti:

- Datu ievades lauki:

- Mehāniskais skaitītājs "In"
- Mehāniskais skaitītājs "Out"
- Elektroniskais skaitītājs "In"
- Elektroniskais skaitītājs "Out"
- Poga "Ok", kas, pirmo reizi nospiežot, saglabā ievadītās vērtības un, otro reizi nospiežot, strādā kā saite uz **Naudas uzpilde** skatu
- Poga "El/Meh", kas pārslēdz skaitītāju datu ievadi no elektroniskā uz mehānisko skaitītāju vai otrādi
- Poga "Labot", kas ļauj izlabot ievadītās vērtības
- Poga "Tālāk", kas ir saite uz **Naudas uzpilde** skatu
- Inkasācijas skaitītāju datu ievades rezultāts (parādās tikai pēc skaitītāju datu ievades vai labojot jau esošos skaitītāju datus):
 - Skaitītāju ievades dati
 - Iespēlētie kredītpunkti
 - Likme
 - Ieņēmums

2.5.9.2. Naudas uzpilde

Joker

LA-75243

Viltojumi (Eur):
123

Uzpilde (Eur):
1

Samazināšana (Eur):
123

Demontāža

232138.00

Izņemts pamatkapitāls 700.00 (Eur)

Ok Atcelt

19. attēls, Skaitītāju ievades skats

Šajā skatā papildus rīkjoslai tiek attēlots (pēc noklusētās valūtas):

- Viltojumu daudzums
- Automāta uzpildes daudzums inkasācijas laikā
- No automāta izņemtais naudas daudzums inkasācijas laikā
- Vai automāts tiek demontēts

2.5.10. IA numuru saraksts

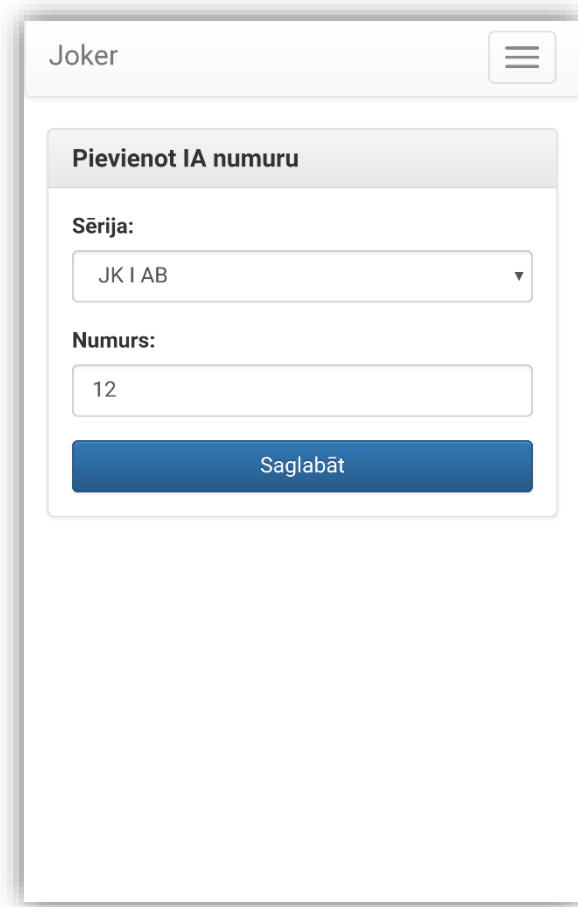


20. attēls, IA numuru saraksta skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Inkasācijas aktu ievadīto numuru saraksts ar diviem tipiem – “Anulēts” vai “Izlietots”.
- Poga “Pievienot numuru”, kas ir saite uz **IA numura pievienošana** skatu
- Poga “Anulēt numuru”, kas ir saite uz **IA numura anulēšana** skatu

2.5.11. IA numura pievienošana



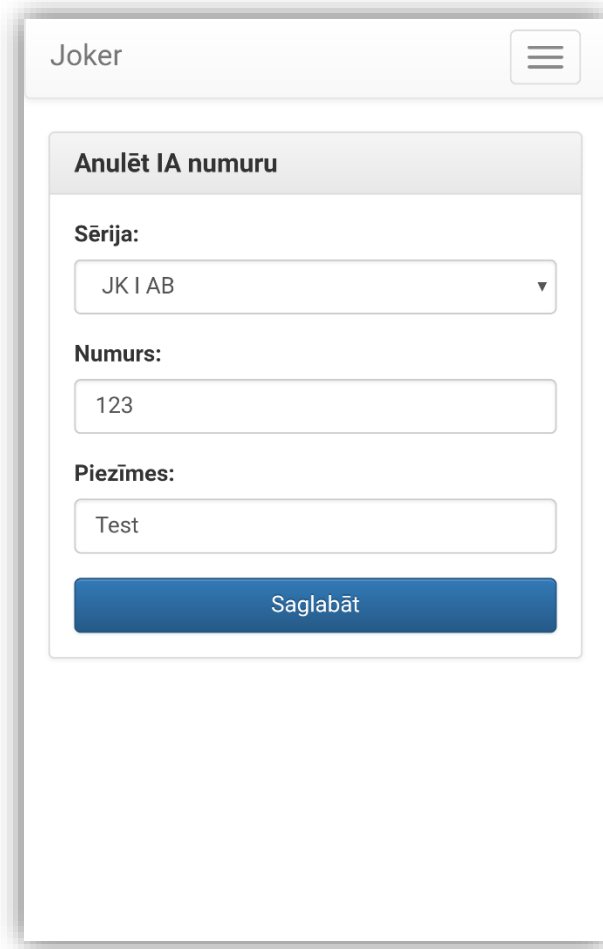
The screenshot shows a mobile application interface for adding an IA number. At the top, the title 'Joker' is displayed on the left and a hamburger menu icon on the right. Below this is a form titled 'Pievienot IA numuru'. The form contains two input fields: 'Sērija:' with a dropdown menu showing 'JK I AB' and a downward arrow, and 'Numurs:' with a text input field containing '12'. At the bottom of the form is a blue button labeled 'Saglabāt'.

21. attēls, IA numura pievienošanas skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Inkasācijas akta numura ievades lauki:
 - Sērija
 - Numurs
- Poga “Saglabāt”, kas saglabā ievadītos datus
- Paziņojums, ja tiek nepareizi ievadīti dati, kas parādās pēc datu saglabāšanas

2.5.12. IA numura anulēšana



The screenshot shows a mobile application interface for 'Joker'. At the top, there is a header with the text 'Joker' and a hamburger menu icon. Below the header is a form titled 'Anulēt IA numuru'. The form contains three input fields: a dropdown menu for 'Sērija' with the value 'JK I AB', a text input field for 'Numurs' with the value '123', and a text input field for 'Piezīmes' with the value 'Test'. At the bottom of the form is a blue button labeled 'Saglabāt'.

22. attēls, IA numura anulēšanas skats

Šajā skatā papildus rīkjoslai tiek attēlots:

- Inkasācijas akta numura ievades lauki:
 - Sērija
 - Numurs
 - Piezīmes
- Poga “Saglabāt”, kas saglabā ievadītos datus
- Paziņojums, ja tiek nepareizi ievadīti dati, kas parādās pēc datu saglabāšanas

2.5.13. Par lietotni



13. attēls, Par lietotni skats

Šajā skatā tiek attēlots:

- Lietotnes izstrādātāja logo
- Pašreizējā lietotnes versija un versijas izstrādāšanas datums
- Lietotnes izstrādātāja nosaukums

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

Šajā nodaļā tiek aprakstīti inkasācijas sistēmas lietotnes testēšanas rezultāti. Testēšanā tika izmantots “melnās kastes” princips, kur tiek testēta programmatūras funkcionalitāte, neskatoties uz iekšējo programmatūras struktūru. Papildus tam izstrādes laikā tika izmantota arī vienībtestēšana, ko nodrošina *Jest* ietvars (sīkāk skat. **Pielikums - Automatizēto vienībtestu rezultātu piemēri**).

3.2. Testējamās raksturiezīmes

Tiek noskaidrots, vai autentifikācijas dati tiek pareizi saglabāti, kā arī tiek pārbaudīts, vai iespējams izmainīt saglabātos datus. Sinhronizācijas procesā, tiek pārbaudīts, vai tiek parādīts paziņojums par kļūdainiem autentifikācijas datiem, vai pēc datu lejupielādes visi saņemtie dati tiek pareizi ievietoti datubāzē, vai tiek veiksmīgi augšupielādēti ievadītie inkasācijas dati, vai iespējams izdzēst nepareizi ievadītos datus.

Inkasācijas datu pārvaldības kontekstā tiek pārbaudīts, vai adresu, instalāciju un to saistītā informācija tiek korekti attēlota, vai pareizi tiek izvēlēts noskenētais svītrkods, vai ievadītie inkasācijas dati tiek pareizi saglabāti un attēloti, vai labojot un izdzēšot inkasācijas datus netiek atgriezti kļūdaini rezultāti.

Inkasācijas aktu pārvaldības kontekstā tiek noskaidrots, vai informācija inkasācijas akta izdrukā tiek attēlota pareizi, vai ievadītie inkasācijas aktu numuru korekti uzrādās inkasācijas aktu numuru sarakstā, vai nepareizas numuru ievades gadījumā tiek izvadīti attiecīgie kļūdu paziņojumi.

Tika pārbaudīts, vai lietotne izmanto SSL savienojumu ar serveri, vai operētājsistēmu minimālās versijas atbilst nefunkcionālajās prasībās norādītajam, kā arī tika pārbaudīts, vai iespējams ievadīt inkasācijas datus bezsaistē, vai sinhronizāciju iespējams veikt izmantojot gan WiFi, gan mobilo internetu.

Testēšanas laikā dažādās situācijās tiek testēti visi lietotāju stāsti un nefunkcionālās prasības.

3.3. Testpiemēru specifikācija

Testi apraksts tiek attēlots tabulas formā izmantojot sekojošu struktūru:

- Nr.p.k – numurs pēc kārtas
- Testa nosaukums

- Ievaddati
- Sagaidāmais rezultāts
- Rezultāts ('+' – ja testa rezultāts sakrīt ar sagaidāmo rezultātu, citādi tiek aprakstīts kļūdas iemesls)
- Lietotāju stāsts, kura kontekstā tiek izpildīts konkrētais tests

Pārskatāmības dēļ, funkcionālo prasību testi tiek iedalīti 3 moduļos – tāpat kā tas tiek attēlots programmatūras projektējuma aprakstā – **2. līmenis - viedtālrunu lietotne**(Datu sinhronizācija, Inkasācijas datu pārvaldība, inkasācijas aktu pārvaldība). Testpiemēri, kuri iepriekš devuši pozitīvu rezultātu, nākošajās testēšanas reizēs vairs netiek uzrādīti.

3.3.1. Funkcionālās prasības

3.3.1.1. Datu sinhronizācija

Pirmā testēšanas reize:

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Autentifikācijas datu saglabāšana	Lietotājvārds un parole	Autentifikācijas dati tiek saglabāti datubāzē	+	1897
2.	Autentifikācijas datu izmainīšana	Lietotājvārds un parole	Datubāzē tiek izmainīti esošie autentifikācijas dati pret jaunajiem	+	1897
3.	Sinhronizācijas datu lejupielāde	Autentifikācijas dati	Visi saņemtie dati tiek veiksmīgi ievietoti datubāzē	+	1898
4.	Sinhronizācijas datu augšupielāde	Inkasācijas dati un aktu numuri	Ievadītie dati tiek veiksmīgi augšupielādēti uz servera un saņemtie	Augšupielādes dati tika nepareizi ievietoti XML failā	1898

			identifikatori saglabāti datubāzē		
5.	Kļūdainu autentifikācijas datu paziņojums	Kļūdaini autentifikācijas dati	Tiek parādīts paziņojums par kļūdainiem autentifikācijas datiem un dota iespēja izlabot tos	+	1898
6.	Kļūdainu sinhronizācijas datu fiksēšana	Kļūdaini inkasācijas dati	Tiek parādīts paziņojums par kļūdainajiem ierakstiem un dota iespēja izlabot ievadītos datus.	+	1898
7.	Sinhronizācijas datu labošana	Izvēlētie inkasācijas dati	Visi izvēlētie inkasācijas dati tiek izdzēsti	+	1898

27. tabula, Datu sinhronizācijas pirmās testēšanas reizes rezultāti

Otrā testēšanas reize (tiek atkārtoti testi, kas nav devuši pozitīvus rezultātus pirmajā reizē):

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Sinhronizācijas datu augšupielāde	Inkasācijas dati un aktu numuri	Ievadītie dati tiek veiksmīgi augšupielādēti uz servera un saņemtie identifikatori saglabāti datubāzē	+	1898

28. tabula, Datu sinhronizācijas otrās testēšanas reizes rezultāti

3.3.1.2. Inkasācijas datu pārvaldība

Pirmā testēšanas reize:

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Inkasācijas paneļa datu pareizības pārbaude	Sinhronizācijas lejupielādes dati	Tiek pareizi attēlots lietotāja vārds, uzvārds, viņam piešķirto adresu un instalāciju skaits	+	1905
2.	Adrešu saraksta pareizības pārbaude	Sinhronizācijas lejupielādes dati	Tiek pareizi attēlots inkasentam piešķirto adrešu saraksts	+	1905
3.	Spēļu automātu saraksta pareizības pārbaude	Adrešu identifikators	Tiek pareizi attēlots inkasentam piešķirto spēļu automātu saraksts	+	1905
4.	Spēļu automāta svītrkoda skenēšanas pārbaude	Adrešu identifikators	Noskenējot instalācijas svītrkodu, tiek atvērts inkasācijas datu ievades skats noskenētajai instalācijai	+	1905
5.	Inkasācijas ieraksta izveide	Instalācijas identifikators, mehānisko un elektronisko skaitītāju dati	Ievadītie dati tiek parādīti tabulā, kopā ar iepriekš ievadītajiem datiem un ieņēmumiem, iespēļētajiem kredītpunktiem un likmi	Iepriekš ievadītie dati netiek parādīti un nestrādā pārslēģšanās starp mehānisko un elektronisko skaitītāju	1916

6.	Inkasācijas ieraksta labošana	Instalācijas identifikators, mehānisko un elektronisko skaitītāju dati	Labotie dati tiek parādīti tabulā kopā ar iepriekš ievadītajiem datiem un ieņēmumiem, iespēļētajiem kredītpunktiem un likmi.	Netika izpildīts, jo “Inkasācijas ieraksta izveide” testa rezultāts ir kļūdainš	1916
7.	Inkasācijas ieraksta dzēšana	Instalācijas identifikators	Inkasācijas ieraksts vairs neparādās instalāciju sarakstā	+	1916

29. tabula, Inkasācijas datu pārvaldības pirmās testēšanas rezultāti

Otrā testēšanas reize (tiek atkārtoti testi, kas nav devuši pozitīvus rezultātus pirmajā reizē un kas netika izpildīti):

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Inkasācijas ieraksta izveide	Instalācijas identifikators, mehānisko un elektronisko skaitītāju dati	Ievadītie dati tiek parādīti tabulā, kopā ar iepriekš ievadītajiem datiem un ieņēmumiem, iespēļētajiem kredītpunktiem un likmi	+	1916
2.	Inkasācijas ieraksta labošana	Instalācijas identifikators, mehānisko un elektronisko skaitītāju dati	Labotie dati tiek parādīti tabulā kopā ar iepriekš ievadītajiem datiem un ieņēmumiem, iespēļētajiem kredītpunktiem un likmi.	+	1916

30. tabula, Inkasācijas datu pārvaldības otrās testēšanas rezultāti

3.3.1.3. Inkasācijas aktu pārvaldība

Pirmā testēšanas reize:

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Inkasācijas akta izdrukas pareizības pārbaude	Konkrētās adreses visi ievadītie inkasācijas dati	Inkasācijas akta izdruka izskatās tieši tāpat, kā tā redzama pielikumā – Inkasācijas akta izdrukas paraugs	Inkasāciju sarakstā tiek iekļauti “Jackpot” inkasācijas ieraksti	1918
2.	Inkasācijas aktu numuru saraksts	Konkrētās adreses visi piesaistītie un anulētie inkasācijas aktu numuri	Piesaistītie un anulētie inkasācijas aktu numuri tiek attēloti saraksta veidā ar numuru, tipu un statuss - izlietots vai anulēts	+	1919
3.	Inkasācijas akta numura ievades kļūdas attēlošana, ja numurs jau izlietots	Sērija, numurs	Kļūdas paziņojums par to, ka inkasācijas numurs jau ticis izlietots norādītajā datumā	+	1920, 1921
4.	Inkasācijas akta numura ievades kļūdas attēlošana, ja numurs nav piešķirts	Sērija, numurs	Kļūdas paziņojums par to, ka inkasācijas numurs nav piešķirts konkrētajam darbiniekam	+	1920, 1921

5.	Inkasācijas akta numura ievade	Sērija, numurs	Inkasācijas numurs tiek pievienots inkasācijas aktu numuru sarakstam	+	1920, 1921
----	--------------------------------	----------------	--	---	------------

31. tabula, Inkasācijas aktu pārvaldības pirmās testēšanas rezultāti

Otrā testēšanas reize (tiek atkārtoti testi, kas nav devuši pozitīvus rezultātus pirmajā reizē):

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Inkasācijas akta izdrukas pareizības pārbaude	Konkrētās adreses visi ievadītie inkasācijas dati	Inkasācijas akta izdruka izskatās tieši tāpat, kā tā redzama pielikumā – Inkasācijas akta izdrukas paraugs	Inkasāciju kopsavilkumā netiek ierēķināti “Jackpot” inkasāciju rādījumi	1918

32. tabula, Inkasācijas aktu pārvaldības otrās testēšanas rezultāti

Trešā testēšanas reize (tiek atkārtoti testi, kas nav devuši pozitīvus rezultātus otrajā reizē):

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Lietotāju stāsts
1.	Inkasācijas akta izdrukas pareizības pārbaude	Konkrētās adreses visi ievadītie inkasācijas dati	Inkasācijas akta izdruka izskatās tieši tāpat, kā tā redzama pielikumā – Inkasācijas akta izdrukas paraugs	+	1918

33. tabula, Inkasācijas aktu pārvaldības trešās testēšanas rezultāti

3.3.2. Nefunkcionālās prasības

Nr.p.k.	Testa nosaukums	Ievaddati	Sagaidāmais rezultāts	Rezultāts	Identifikators
1.	Tīkla izmantošanas pārbaude	Inkasācijas dati	Sinhronizācija iespējama gan caur mobilo, gan <i>WiFi</i> tīklu	+	NP1001
2.	Atbalstīto versiju pārbaude	Nav	Lietotne strādā uz Android 4.4 un lielākas versijas, kā arī uz iOS 8.0 un lielākas versijas	+	NP1002
3.	Droša savienojuma pārbaude	Inkasācijas dati	Dati tiek saņemti un nosūtīti caur <i>HTTPS</i> savienojumu	+	NP1003

34. tabula, nefunkcionālo prasību rezultāti

4. PROJEKTA ORGANIZĀCIJA

Projekts tika izstrādāts, izmantojot spējas programmatūras izstrādes pieeju, lai nodrošinātu pēc iespējas labāku atgriezenisko saiti ar klientu un laicīgāk pamanītu nepilnības, ko klients vēlas, lai izstrādātājs izlabo.

Spējās programmatūras izstrādes laikā tika izmantots SCRUM modelis, kas sastāv no 2 nedēļu gariem sprintiem, kura laikā tika izpildītas konkrētajam sprintam uzdotās prasības. Katru darba dienu ap pusdienlaiku notika sapulce, kurā tika apspriests, kas pagājušajā dienā ir paveikts, kas būtu jā dara turpmāk. Šī projekta ietvaros SCRUM vadītājs ir tas pats cilvēks, kas vada projekta izstrādi (projekta vadītājs).

Visas programmatūras prasības/darba uzdevumi tika sadalīti lietotāju stāstos iekš Redmine darbu pārvaldības sistēmas. Katram lietotāju stāstam projekta vadītājs iedala paredzēto laiku, cik ilgi varētu notikt izstrādes process. Ja programmatūras izstrādes procesā radušās kādas kļūdas, tad tiek izveidots jauns ieraksts Redmine sistēmā kā "Bug", kas, visbiežāk, ir jāizlabo programmētājam, kurš izstrādes procesā ir radījis šo kļūdu.



14. attēls, SCRUM modelis[7]

5. KVALITĀTES NODROŠINĀŠANA

Izstrādātās lietotnes kvalitātes nodrošināšanai, kvalifikācijas darba autors veidoja programmatūras dokumentāciju, kurā tika noteiktas visas programmatūras prasības, projektējums, ka arī testēšanas dokumentācija.

Programmatūras prasību definēšanai netika izmantotas valsts standartos noteiktās vadlīnijas, jo prasību aprakstīšanai tika izmantota spējā programmatūras izstrādes pieeja. Programmatūras testēšanas dokumentācija un projektējums tika veidots vadoties pēc sekojošiem valsts standartiem programminženierijā:

- LVS 70:1996 “Programmatūras testēšanas dokumentācija”
- LVS 72:1996 “Ieteicamā prakse programmatūras projektējuma aprakstīšanai”

Koda kvalitātes nodrošināšanai kvalifikācijas darba autors ievēroja Javascript programmēšanas valodas sintakses īpašības un labo praksi[6]. Pēc katra lietotāja stāsta izstrādes, darba autors veica koda refaktorēšanu, kā arī vienībtestus, izmantojot *Jest* ietvaru, un akcepttestus katram izstrādātajam lietotāju stāstam atsevišķi, lai nodrošinātu pareizu programmatūras darbību.

Pēc veiksmīgi izietiem testiem katram lietotāju stāstam projekta/darba vadītājs pārskatīja manis rakstīto kodu GitLab versiju kontroles sistēmas mājaslapā un, ja, viņaprāt, nebija pilnībā/pareizi izpildītas funkcionālas prasības vai netika ievērota labā programmēšanas prakse, darba vadītājs, komentējot, lika pārrakstīt kļūdainās vietas kodā pirms tā savienošanas ar galveno izstrādes zaru.

Sakarā ar to, ka React-Native ir starp-platformu ietvars, lietotne tika periodiski notestēta uz fiziskiem Android un iOS operētājsistēmu viedtālruniem, lai nodrošinātu programmatūras kvalitāti uz dažādām platformām.

6. KONFIGURĀCIJU PĀRVALDĪBA

Izstrādājot kvalifikācijas darbā aprakstīto programmatūru, tika izmantots *GitLab* kā *Git* repositorijs un koda pārvaldības rīks. Izmantojot *Git*, ir iespējams kontrolēt programmatūras koda un versiju izmaiņas, kā arī koordinēt programmatūras izstrādes darbu starp vairākiem izstrādātājiem vienlaicīgi.

Kvalifikācijas darba ietvaros katram lietotāju stāstam tika izveidots atsevišķs izstrādes zars, lai kods būtu viegli trasējams. Pēc katra zara izstrādes, projekta vadītājs pārskatīja uzrakstīto kodu un, pēc pozitīvas pārskates, savienoja to ar galveno izstrādes zaru, no kura tika būvētas lietotnes versijas priekš Android un *iOS* operētājsistēmām. Katrai būvējuma versijai tika piešķirts savs tags - t.i. *Jenkins* būvējuma numuram atbilstoši tiek pielikts Git tags un pielikta lietotnes versija, kas parādās iekš *Fabric* (sīkāk skat. **Pielikumi – Konfigurācijas pārvaldība**).

Fabric ir mobilā platforma, kas palīdz izstrādātājiem:

- Analizēt lietotnes lietošanas procesā radušās kļūdas.
- Pārskatīt ieņēmumus, kas ir radušies no lietotnes.
- Analizēt statistiku, par to cik daudz un cik bieži lietotāji lieto izstrādāto lietotni.
- Nodrošināt izstrādātājus ar platformu, kurā var privāti publicēt savas lietotnes (kvalifikācijas darba gadījumā, lai tikai noteikti inkasenti varētu lietot inkasācijas sistēmas viedtālrunu lietotni).[8]

7. DARBIETILPĪBAS NOVĒRTEJUMS

Sākot darbu pie lietotnes, projekta vadītājs iedalīja katram lietotāju stāstam atvēlēto laiku – darbietilpības novērtējumu, tai pašā laikā kvalifikācijas darba autors, skatoties pēc savas iepriekš gūtās pieredzes, deva norādes projekta vadītājam, cik katra lietotāju prasības izstrāde varētu aizņemt laiku. Veicot darbietilpības novērtējumu, viens funkcijs punkts tika pielīdzināts vienai cilvēkdienai, kas ir 8 stundas.

Sekojošā tabula attēlo, kāds katram lietotāju stāstam ir novērtētais un reālais funkcijs punktu apjoms:

Lietotāja stāsta identifikators	Darbietilpības novērtējums(funkcijs punkti)	Reālā darbietilpība (funkcijs punkti)
1897	2	3
1898	16	20
1905	2	2
1906	4	3
1915	5	5
1916	5	5
1917	4	3
1918	7	9
1919	5	5
1920	4	5
1921	4	2
Kopā:	58	62

Saskaitot kopā visus lietotāju stāstu darbietilpības novērtējumus, nepieciešamais laiks *ReactJS* lietotnes izstrādāšanai tika vērtēts kā 58 cilvēkdienas, taču reālā darbietilpība, kas bija nepieciešama lietotnes izstrādei, ir vērtējama 62 cilvēkdienas apjomā, pārsniedzot prognozēto darbietilpības novērtējumu par 4 cilvēkdienām, kas arī ir aptuveni 3 personmēneši. Papildus lietotāju stāstu izstrādei, bija nepieciešams izstrādāt arī pašu *React-Native* lietotni, kas aizņēma aptuveni 5 cilvēkdienas.

8. PIELIKUMI

8.1. Inkasācijas akta izdrukas paraugs

18.05.2017

SIA "JOKER LTD"

Kairīmas iela 12, Rīga, M-1045

40003266078

Spēļu zāle Aotū 57/59, Rīga

Nr.p.k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Kopā
Autom. Nr	LA-75055	LA-75163	LA-75243	LA-75305	LA-75457	LA-75840	LA-75841	LA-76062	LA-76113	LA-76130	LA-76233	LA-76237	LA-76239	LA-77125	X
Autom. ID.Nr	36872	36872	37227	37763	37974	39209	39849	39849	40172	40189	43349	43353	43355	42854	X
Iepriekš. inkasācijas datums	09.05.2017	09.05.2017	09.05.2017	09.05.2017	09.05.2017	09.05.2017	02.05.2017	08.01.2014	08.01.2014	08.01.2014	08.01.2014	08.01.2014	08.01.2014	08.01.2014	X
Dalības maksa (E) EUR	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	X
Iemaksātās likmes	2151315	0	0	0	0	0	0	760793	506210	628945	199181	140011	167346	236130	X
Iepriekš (A1)	64	0	0	0	44	44	5454	0	0	0	0	0	0	0	X
Iepriekš (A2)	7	7	7	7	9	9	7	7	7	7	7	7	7	7	X
Cip.sk. (A3)	1	0	0	0	1	1	1	1	1	1	1	1	1	1	X
Apil. (A4)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X
Koef. (A5)	100	100	100	100	1	1	100	100	100	100	100	100	100	100	X
Iemaksāts pēc skaitļiem EUR (A6)	784686,00	64,00	0,00	1,00	0,44	0,44	5454,00	9239207,00	9493790,00	9371055,00	9800819,00	9859989,00	9832654,00	9763870,00	75215589,88
((A2+10*A3+A4)*A5+E)	5134346	0	0	0	0	0	0	695689	456606	581067	185421	129303	155859	221358	X
Izmaksātās likmes	2	4	0	2	4	444	0	0	0	0	0	0	0	0	X
Iepriekš (B1)	7	7	7	7	9	9	7	7	7	7	7	7	7	7	X
Iepriekš (B2)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X
Cip.sk. (B3)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X
Apil. (B4)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X
Koef. (B5)	100	100	100	100	1	1	100	100	100	100	100	100	100	100	X
Izmaksāts pēc skaitļiem EUR (B6)	4865656,00	4,00	0,00	2,00	0,04	4,44	0,00	9304311,00	9545394,00	9418933,00	9814579,00	9870697,00	9844141,00	9778642,00	72440363,48
((B2+10*B3+B4)*B5+E)															X
Izm. likmes ar krat.															X
/no kasas pēc skait.															X
Iepriekš (C1)															X
Iepriekš (C2)															X
Cip.sk. (C3)															X
Apil. (C4)															X
Koef. (C5)															X
Izmaksāts pēc skaitļiem EUR (C6)															X
((C2+10*C3+C4)*C5+E)															X
Kopā nopelnīts pēc skaitļiem EUR	2983030,00	60,00	0,00	-1,00	0,40	-4,00	5454,00	-65104,00	-49604,00	-47878,00	-13760,00	-10708,00	-11487,00	-14772,00	2775226,40
A6-(B6+C6)															0,00

3.Tabula. Inkasācija

Nr.p.k	Maksājuma izziņis	Summa cipars	Latvijas Bankas kurss	Summa EUR
1	EUR			2775226,40
2	USD			
3	Norēķinu kartes			
Kopā	X	X		2775226,40

- Izmaksāts uzdevātais Džekpots: -568454,00 (EUR)
- Negatīvais ienākums no naučādi pielīdzinātajiem priekšmielēm: 12,00 (EUR)
- Azartspēļu rezultāts (1,-2,+3): 3323668,40 (EUR)

Akta sastādītājs:

(vārds, uzvārds, amats)

Akta pieņēmējs:

(vārds, uzvārds, amats)

8.2. Automatizēto vienībtestu rezultātu piemēri

```
FAIL src\__tests__\Utilities.test.js
? Utilities > Calculate income/outcome for encashment

expect(received).toBe(expected)

Expected value to be (using ===):
  25524
Received:
  255.24

    at Object.<anonymous> (src\__tests__\Utilities.test.js:61:24)
    at process._tickCallback (internal\process\next_tick.js:109:7)

Utilities
  ✓ Fills array with empty values + number property for printout (1ms)
  ✓ Calculate rounds for encashment (1ms)
  ✗ Calculate income/outcome for encashment (1ms)

PASS src\__tests__\EncashmentsService.test.js
EncashmentsService
  ✓ Save Encashment (32ms)
  ✓ Get lastest encashments by statement (3ms)

PASS src\__tests__\DataSyncService.test.js
DataSyncService
  ✓ Build XML (33ms)
  ✓ Parse XML (41ms)
  ✓ Translate ERROR codes (5ms)
  ✓ Get column types from Database Schema (2ms)

PASS src\__tests__\EmployeeService.test.js
EmployeeService
  ✓ Get Employee (40ms)

PASS src\__tests__\StatementsService.test.js
StatementsService
  ✓ Save statement (57ms)
  ✓ Get unsynced statements (3ms)

PASS src\__tests__\StatementTemplateService.test.js
StatementTemplateService
  ✓ Save TemplateUsage (21ms)
  ✓ Get TemplateUsages (2ms)

PASS src\__tests__\ApplicationSettingsService.test.js
ApplicationSettingsService
  ✓ Authentication data save (22ms)
  ✓ LastUploadTime save (1ms)

PASS src\__tests__\DbManager.test.js
DbManager
  ✓ GetConnection returns valid connection (21ms)
  ✓ GetConnection executes all callbacks (4ms)

Test Suites: 1 failed, 7 passed, 8 total
Tests: 1 failed, 17 passed, 18 total
Snapshots: 0 total
Time: 1.101s
Ran all test suites.
```

8.3. Konfigurācijas pārvaldības

8.3.1. Git

Graph	Description	Date	Author	Commit
	#2132	15 Maijs 2017 10:45	Toms Muceniaks <	d74024c
	Merge branch 'master' into feature/2132	13 Maijs 2017 16:27	Toms Muceniaks <	c4154d5
	Merge branch 'feature/2130' into 'master'	13 Maijs 2017 16:19	Toms Muceniaks <	013689d
	origin/feature/2130 #2130	13 Maijs 2017 16:18	Toms Muceniaks <	361804d
	Merge branch 'feature/2130' into 'master'	13 Maijs 2017 16:05	Kalvis Upitis <kalvi	425b150
	#2130[MERGING	13 Maijs 2017 16:03	Toms Muceniaks <	90ec3ab
	#2130	13 Maijs 2017 15:52	Toms Muceniaks <	1ee5691
	#2132	13 Maijs 2017 15:19	Toms Muceniaks <	b57c85c
	ios-37 Merge branch 'feature/2119' into 'master'	13 Maijs 2017 15:12	Kalvis Upitis <kalvi	b42cc0b
	#2130	13 Maijs 2017 15:01	Toms Muceniaks <	e5119fb
	origin/feature/2119 #2119	13 Maijs 2017 14:42	Toms Muceniaks <	ac92d98
	#2119	13 Maijs 2017 11:54	Jenkins Slave <jenl	8f94481
	#2119	13 Maijs 2017 10:56	Toms Muceniaks <	01e9ef8
	#2130	13 Maijs 2017 10:05	Toms Muceniaks <	6e2c109
	#2130	12 Maijs 2017 14:01	Toms Muceniaks <	9943275
	#2130	11 Maijs 2017 15:13	Toms Muceniaks <	b87c1bb
	#2130	10 Maijs 2017 17:41	Toms Muceniaks <	ecbe274

8.3.2. Jenkins

Jenkins

Jenkins > react-native-ios > #37

- Back to Project
- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete Build
- Environment Variables
- Git Build Data
- No Tags
- Previous Build
- Next Build

Build #37 (May 13, 2017 3:17:36 PM)

Build Artifacts

- JokerReactNative.app.dSYM.zip 864.74 KB [view](#)
- JokerReactNative.ipa 4.42 MB [view](#)

Changes

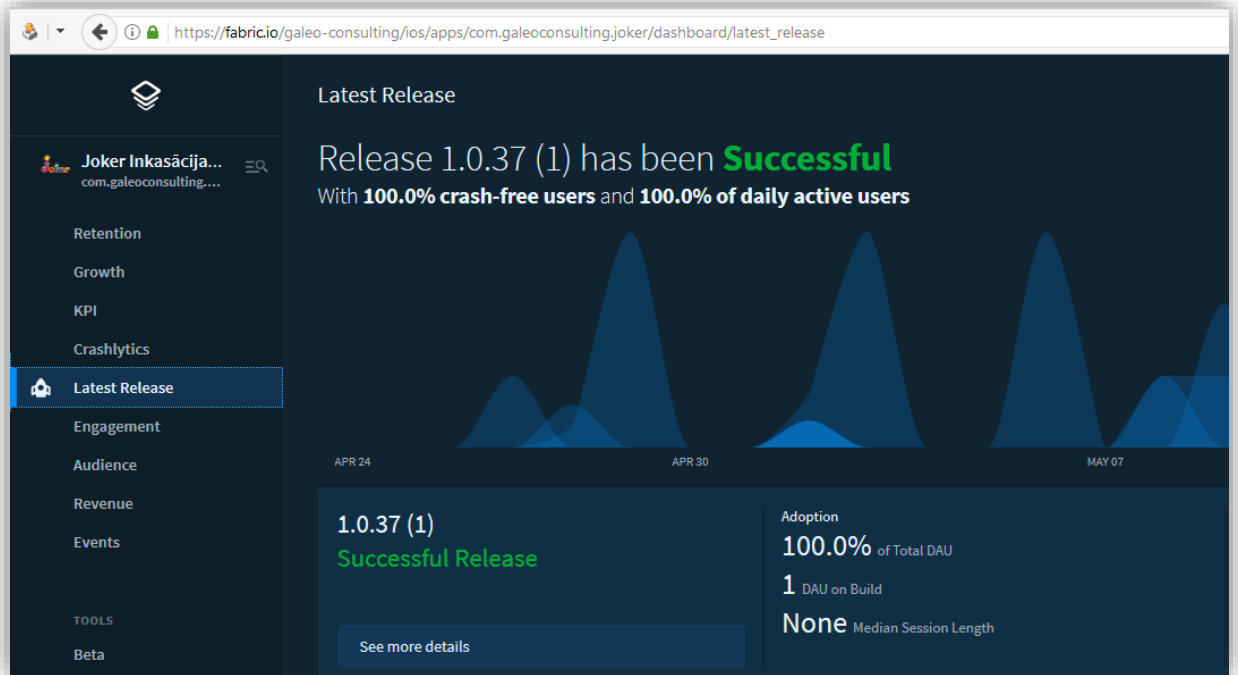
- publish profile ([detail](#))
- Fixed summer and winter month indexes ([detail](#))
- removed file added by accident ([detail](#))
- #2119 ([detail](#))
- #2119 ([detail](#))
- #2119 ([detail](#))

Started by user [Kalvis Upitis](#)

Revision: b42cc0b019c3ffc8caed93fb28e95c10fd9d2775

- origin/master

8.3.3. Fabric



8.4. Koda fragmenti

8.4.1. Datu sinhronizācijas skats

```
class DataSync extends Component {
  constructor() {
    super();
    this.state = {
      progressValue: 0,
      progressState: "Nospiediet pogu Sinhronizēt.",
      editData: false,
      wrongCredentials: false
    }
    this.handleSubmit = this.handleSubmit.bind(this);
    this.handleDataEdit = this.handleDataEdit.bind(this);
    this.handleCredentialLink = this.handleCredentialLink.bind(this);
  }
  handleSubmit() {
    var dataSyncService = new DataSyncService();
    //Populates view with response from Synchronization.
    dataSyncService.startDataSync(function (dataStatus) {
      this.setState({
        progressValue: dataStatus.loading,
        progressState: dataStatus.status,
        editData: dataStatus.editData,
        wrongCredentials: dataStatus.wrongCredentials
      });
    }).bind(this);
  }
  handleDataEdit() {
    browserHistory.push('/edit_syncable_data/'+this.state.progressState);
  }

  handleCredentialLink(){
```

```

    browserHistory.push('/authentication');
  }
  render() {
    return (
      <div>
        <div className="panel panel-default">
          <div className="panel-heading">
            <h3 className="panel-title"><strong>Datu sinhronizācija</strong></h3>
          </div>
          <div className="panel-body">
            <button type="button"
              className="btn btn-primary btn-block"
              data-toggle="modal" data-target="#startSync">Sinhronizēt</button>
            {this.state.editData &&
              <button
                type="button"
                className="btn btn-primary btn-block"
                onClick={this.handleDataEdit}>Rediģēt sinhr. datus</button>
            }
            {this.state.wrongCredentials &&
              <button
                type="button"
                className="btn btn-primary btn-block"
                onClick={this.handleCredentialLink}>Nomainīt autentifikācijas
datus</button>
            }
            {!(this.state.progressValue === 0 || this.state.progressValue === 100)
&&
              <div className="progress">
                <div
                  className="progress-bar progress-bar-striped active"
                  role="progressbar"
                  style={{ width: this.state.progressValue + '%' }} />

```

```

        </div>
    }
    <strong>Progress:</strong> {this.state.wrongCredentials ?
"Nepareizautentifikācijas dati!" : this.state.progressState}
    </div>
</div>
<DataSyncStartDialog
    handleSubmit={this.handleSubmit}
/>
</div>
);
}
}
export default DataSync;

```

8.4.2. XML faila sagatavošana datu nosūtīšanai

Kvalifikācijas darbā neietilpst koda daļas, kas saistītas ar “Work”!

```

buildXML(encashmentData, workData) {
  //Template for push.xml
  var dataForXML = {
    PushRequest: [
      {
        Timestamp: Date.now()
      },
      {
        Create: [
          { Statements: [] },
          { Encashments: [] },
          { StatementTemplateUsages: [] },
          { WorkDays: [] },
          { Works: [] },
          { WorkDayMoneyMovements: [] }

```

```

        ]
    },
    {
        Update: [
            { WorkDays: [] },
            { Works: [] },
            { WorkDayMoneyMovements: [] }
        ]
    }
]
};
var dataSyncUtils = new DataSyncUtils();

if (encashmentData) {
    //Write data for unsynced encashments
    dataSyncUtils.pushEncashments(encashmentData, dataForXML);
    //Write data for unsynced statements
    dataSyncUtils.pushStatements(encashmentData, dataForXML);
    //Write data for unsynced statementTemplateUsages
    dataSyncUtils.pushStatementTemplateUsages(encashmentData, dataForXML);
}

if (workData) {
    //Write data for unsynced or updated workDays
    dataSyncUtils.pushWorkDays(workData, dataForXML);
    //Write data for unsynced or updated Works
    dataSyncUtils.pushWorks(workData, dataForXML);
    //Write data for unsynced or updated MoneyMovements
    dataSyncUtils.pushMoneyMovements(workData, dataForXML);
}
var xml = builder.create(dataForXML, { separateArrayItems: true }).end();
return xml;
}

```

9. SECINĀJUMI

Kvalifikācijas darba ietvaros tika izstrādāta inkasācijas sistēmas lietotne, kuru ir iespējams uzstādīt gan uz *Android*, gan *iOS*. Lietotnes izstrādē tika izmantots *React-Native* ietvars, lai arī tas ļāva izstrādāt lietotni lielākoties uz abām operētājsistēmām vienlaicīgi, tomēr dažkārt nācās rakstīt platformai specifisku kodu, it īpaši failu sistēmas operācijām. Savukārt, *ReactJS* ļāva norobežoties no dažādām negatīvām starp-platformu programmatūras izstrādes iezīmēm: atviegloja saskarnes izstrādi, paātrināja testēšanas procesu un atklūdošanu, jo nebija nepieciešams izvietot lietotni uz emulatora vai fiziska viedtālruna, kas, savukārt, ļāva lietotnes pamata funkcionalitāti testēt interneta pārlūkprogrammā.

Uzrakstītajai programmatūrai tika izstrādāta atbilstoša dokumentācija, kur tiek aprakstītas visas prasības, ko klients vēlas redzēt lietotnē, kā arī programmatūras projektējuma apraksts un testi, t. sk. automatizētā vienībtestēšana.

Kvalifikācijas darba izstrādes laikā tika izmantota spējās izstrādes metodoloģija un tika secināts, ka, izmantojot šo pieeju, ir nepieciešams būt precīzam, ar spējām paskaidrot klientam darba progresu tā, lai cilvēks, kas nav programmētājs, saprastu bez liekiem pārpratumiem un jautājumiem, kā arī jābūt gatavam ātri un efektīvi apgūt jaunas tehnoloģijas

Izstrādājot kvalifikācijas darbu, tika apgūtas un/vai izmantotas sekojošas valodas/tehnoloģijas:

- *Javascript*
- *React-Native* un *ReactJS* ietvari
- *JSON*
- *XML*
- *Git* versiju kontroles rīks
- *HTTP* pieprasījumi
- *Jenkins* būvēšanas rīks

Kvalifikācijas darba izstrādes process ir devis neatsveramu pieredzi augstākminētajās tehnoloģijās, taču, bez tām, tika iegūtas prasmes rakstīt programmatūras pavadošo dokumentāciju, veikt darbietilpības novērtējumus, veidot kvalitatīvu, citiem viegli uztveramu, kodu, kā arī strādāt komandā.

LITERATŪRAS SARAKSTS

1. Akadēmiskā terminu datu bāze [tiešsaiste] – [atsauce 29.03.2017.]. Pieejams:
<http://termini.lza.lv/>
2. Windows CE 5.0 atbalsta serviss [tiešsaiste] – [atsauce 25.05.2017] Pieejams:
<https://support.microsoft.com/en-us/lifecycle/search?alpha=Microsoft%20Windows%20CE%205.0>
3. Redmine dokumentācija [tiešsaiste] – [atsauce 14.04.2017] Pieejams:
<http://www.redmine.org/>
4. Likumā noteiktā inkasācijas akta struktūra [tiešsaiste] – [atsauce 23.05.2017] Pieejams:
<https://likumi.lv/doc.php?id=150769>
5. SCRUM – spējās izststrādes ietvars [tiešsaiste] – [atsauce 23.05.2017] Pieejams:
<https://www.scrumalliance.org/>
6. Javascript programmēšanas valodas labā prakse [tiešsaiste] – [atsauce 21.04.2017].
Pieejams: https://www.w3schools.com/js/js_best_practices.asp
7. SCRUM modelis [tiešsaiste] – [atsauce 12.04.2017]. Pieejams:
<http://www.agilenutshell.com/scrum>
8. Fabric dokumentācija [tiešsaiste] – [atsauce 24.05.2017] Pieejams:
<https://docs.fabric.io/>
9. Jenkins dokumentācija [tiešsaiste] – [atsauce .05.2017] Pieejams:
<https://jenkins.io/doc/>

DOKUMENTĀRĀ LAPA

Kvalifikācijas darbs „*Inkasācijas sistēmas uzlabošana*” 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: ***Toms Mucenieks*** _____ ***.05.2017.***

Rekomendēju darbu aizstāvēšanai

Darba vadītājs:

tehniskais projektu vadītājs, M. Dat., Kalvis Upītis _____ ***.05.2017.***

Recenzents: ***IT dienesta vadītājs, M. Dat., Agnis Škuškovniks***

Darbs iesniegts Datorikas fakultātē 29.05.2017.

Dekāna pilnvarotā persona:

docente, Dr. dat. ***Darja Solodovņikova*** _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2017. prot. Nr. _____

Komisijas sekretārs(-e): _____