

Latvijas Universitāte

Datorikas fakultāte

Biroja inventāra uzskaites sistēma

Kvalifikācijas darbs

Autore:

Jelena Fjodorova

St.apl.nr.: jf08049

Vadītājs:

Aivars Niedrītis

M.dat.

RĪGA, 2010

Anotācija

Sistēma „Biroja inventāra uzskaites sistēma” paredzēta ikdienas lietošanai birojā un palīdz pārvaldīt inventāru: apskatīt informāciju par inventāra atrašanās vietu, tā sadalījumu darba vietās un starp darbiniekiem, inventāra vēstures un inventāra pasūtījumiem. Sistēma ir saprotama un ērti lietojama.

Abstract

The Office Supplies Management System is developed for daily use in office in order to manage office supplies: to see information about supplies location, responsible person, work place, history and orders. The system is understandable and easy to use.

IEVADS	6
1 APZĪMĒJUMU UN SAĪSINĀJUMU SARAKSTS	7
2 PROGRAMMATŪRAS PRASĪBU APRAKSTS	8
2.1 IEVADS	8
2.1.1 Dokumenta nolūks.....	8
2.1.2 Dokumenta pārskats	8
2.2 AUGSTĀKĀ LĪMEŅA PRASĪBAS	8
2.3 LIETOTĀJU STĀSTI	9
2.4 LIETOTĀJU AKCEPTESTI	12
2.4.1 Akcepttesti.....	12
3 PROGRAMMATŪRAS PROJEKTĒJUMS	15
3.1 IEVADS	15
3.1.1 Dokumenta nolūks	15
3.1.2 Dokumenta pārskats.....	15
3.2 DATUBĀZES PROJEKTĒJUMS	16
3.2.1 Kopējais ER modelis.....	16
3.2.2 Tabulu apraksts.....	16
3.3 PRASĪBU (LIETOTĀJU STĀSTU) REALIZĀCIJAS UZDEVUMI	24
3.4 ALGORITMU UML AKTIVITĀŠU DIAGRAMMAS.....	29
3.4.1 Jaunu objektu ģenerēšana no pasūtījuma.....	29
3.4.2 Par objektu atbildīgā darbinieka nomaiņšana.....	29
3.4.3 Objekta vēsturisku datu saglabāšana.....	30
4 TESTĒŠANAS DOKUMENTĀCIJA	32
4.1 NOLŪKS	32
4.2 IEVADS	32
4.3 TESTU SAGATAVOŠANA	32
4.4 TESTĒŠANAS REZULTĀTI	32
5 PROJEKTA ORGANIZĀCIJA	36
6 KVALITĀTES NODROŠINĀŠANA	37
7 KONFIGURĀCIJAS PĀRVALDĪBA	38
8 DARBIETILPĪBAS NOVĒRTĒJUMS	39
9 REZULTĀTI	45
10 SECINĀJUMI	46
11 IZMANTOTĀ LITERATŪRA UN AVOTI	47
12 PROGRAMMATŪRAS KODA FRAGMENTI	48
12.1 IEVADS	48
12.2 TABULU SKRIPTI	48
12.3 SECĪBAS (SEQUENCES)	48
12.4 TRIGERI.....	49

12.5	PROCEDŪRAS	51
13	DIAGRAMMU SARAKSTS	52
14	TABULU SARAKSTS	53

Ievads

Lielā uzņēmumā radās nepieciešamība pēs automatizētās inventāra uzskaites sistēmas, lai atvieglotu par inventāra atbildīgās personas darbu. Pamatmērķis: zināt, kur atrodas objekts, kas atbild par to un no kura pasūtījuma tas ir.

Viens no izstrādes nosacījumiem ir: Risinājumam jābūt Oracle bāzētam un ar Oracle Form Builder izstrādātai saskarnei. Sistēmas datubāzei jābūt aizsargātai, tai nedrīkst piekļūt personas, kam nav tādu tiesību. Darbināšanai uz datora jābūt uzstādītai Windows XP OS, Oracle datu bāzei Oracle Database 10.2.0.4 (10.2.0.1 for XP) Enterprise Edition un Oracle Developer Suite 10g 10.1.2.0.2.

Lai sasniegtu darba mērķus, tika nolemts darba autoram veikt šādus uzdevumus:

- iepazīties ar Oracle tehnoloģijām un sagatavot izstrādes vidi
- iepazīties ar līdzšinējo pieredzi līdzīgu sistēmu veidošanā
- iepazīties ar PL/SQL programmēšanas valodas līdzekļiem
- pierakstīt biznesa un funkcionālās prasības un atbilstoši tām veikt projektēšanu
- izstrādāt programmproduktu
- veikt izveidotā programmproduktu vienībtestēšanu un sagatavot visu dokumentāciju

Izstrādājamais programmprodukts ir neatkarīgs un nav lielākas sistēmas sastāvdaļa. Tas ir pilnīgi pielāgots konkrēta uzņēmuma vajadzībām, un tāpēc nav eksistējošās lielākās sistēmas analogs. Produkts ir izstrādāts ar iespēju paplašināt tā darbības sfēru, pielikt jaunas funkcijas. Produkts paredzēts ilgstošai izmantojamībai.

1 Apzīmējumu un saīsinājumu saraksts

Dokumentā tiek izmantoti šādi apzīmējumi:

Jēdziens	Skaidrojums
Adrese	Informācija, kas ir pietiekama, lai atrastu vietu
Inventārs	Visas mēbeles vai iekārtas, kas atrodas birojā
Klasifikators (vai tips)	Nosaka, uz ko attiecas tabulas ieraksts (piem.tabulas Persons ieraksts var aprakstīt ēku(building), darbinieku(employee) vai piegādātāju(deliver) vai nosaka tabulas ieraksta objekta tipu (piem. Tabulas ROOMS klasifikators norāda telpas tipu „meeting room”, „lounge”utt)
Objekts	Dokumentā izmantots kā inventāra apzīmējums
Pavadzīme	Dokuments, kuru izdod juridiska vai fiziska persona - kādas preces piegādātājs
Piegādātāja firma (tekstā arī Piegādātājs)	Inventāra piegādātāja firmas nosaukums vai pati firma
Procedūra	PL/SQL bloks, kas izpilda vienu vai vairākus uzdevumus
Trigeris	PL/SQL struktūra, kas automātiski izpildās, kad tiek izpildīts noteikums, pie kura tas nostrādā (piem. ON INSERT trigeris nostrādās, ja tabulā tiks ievadīti dati)
Vienība	Tekstā domāts inventāra nosaukums vienā pasūtījuma rindā

Tabula 1. Tabula ar izmantotiem apzīmējumiem

Dokumentā tiek izmantoti šādi saīsinājumi:

Saīsinājums	Skaidrojums
BIUS	Biroja inventāra uzskaites sistēma

Tabula 2. Tabula ar izmantotiem saīsinājumiem

2 Programmatūras prasību apraksts

2.1 Ievads

Ievadā tiek apskatītas produkta specifikācijā lietotie saīsinājumi, noteikts dokumenta nolūks. Nodaļa satur arī tālāko nodaļu pārskatu.

2.1.1 Dokumenta nolūks

Dokumenta mērķis ir aprakstīt BIUS funkcionālās un vispārējās prasības, kas ir vajadzīgas, lai uzsāktu risinājuma realizēšanu.

2.1.2 Dokumenta pārskats

1. nodaļa: Ievads Nodaļā ir īsi izklāstīts šī dokumenta un sistēmas, uz kuru dokuments attiecas, mērķis un būtība. Nodaļā sniegts šī dokumenta pārskats.

2. nodaļa: Augstākā līmeņa prasības. Šī nodaļa ietver vai arī satur viena teikuma izklāstītas programmatūras augstākā līmeņa prasības, kas apkopo sistēmas vispārējo biznesa procesu.

3. nodaļa: Lietotāju stāsti Nodaļā satur augstākā līmeņa prasību smalkāko aprakstu lietotāju stāstos, kas veido sistēmas funkcionālas prasības. Katra stāsta numura pirmais cipars sakrīt ar augstākā līmeņa prasības numuru, lai saglabātu prasību trasejumu.

4. nodaļa: Lietotāju akcepttesti. Nodaļā tiek aprakstītas lietotāju akceptēšanas kritēriji pēc lietotāju stāstu realizācijas. Vispārējie akcepttesti ir akcepttesti, kuri ir piemērojami vairāku datu kopām. Katram akcepttestam ir piešķirti lietotāja stāsta vai vairāku numuri, pēc savas būtības tie detalizē kādā lietotāja stāsta aprakstīto funkcionalitāti.

2.2 Augstākā līmeņa prasības

Sistēmas lietotāju administrēšana	
1	Risinājumam jānodrošina iespēja lietotājiem autentificēties un autorizēties sistēmā
Funkcionālās prasības	
2	Risinājumam jānodrošina iespēja uzturēt objektu , telpu, ēku, piegādātāju

	klasifikatorus
3	Risinājumam jānodrošina iespēja manuāli uzturēt darbiniekus
4	Risinājumam jānodrošina iespēja manuāli uzturēt piegādātāju firmas
5	Risinājumam jānodrošina iespēja manuāli uzturēt ēkas
6	Risinājumam jānodrošina iespēja manuāli uzturēt telpas
7	Risinājumam jānodrošina iespēja manuāli uzturēt pieņemtos pasūtījumus
8	Risinājumam jānodrošina iespēja manuāli uzturēt pieņemto pasūtījumu rindas
9	Risinājumam jānodrošina iespēja automatizēti uzturēt objektus
10	Risinājumam jānodrošina iespēja manuāli uzturēt objektus
11	Risinājumam jānodrošina iespēja uzturēt objektu vēsturi
Vispārīgās prasības	
1	Risinājumam jāatbilst spēkā esošiem ārējiem normatīviem aktiem, tai skaitā fizisko personu datu aizsardzības, informācijas drošībā un informācijas pieejamības jomā
2	Risinājumam jābūt pieejamam nepārtraukti
3	Risinājuma katrai darbībai jāizpildās laikā līdz 4 sekundēm
Drošības prasības	
1	Risinājumam jānodrošina datu reserves kopijas veidošanu
2	Risinājumam jānodrošina tikai autorizēta daļa bez publiskas piekļuves

Tabula 3. Tabula ar augstākā līmeņa prasībām

2.3 Lietotāju stāsti

Pārskatāmībai lietotāju stāsti ir attēloti tabulā.

Nr.	Lietotāju stāsti
1.1	Lietotājs var pieslēgties sistēmai ar savu pieteikumvārdu un paroli

1.2	Datu bāzes administrators var pierēģistrēt sistēmas lietotāju datu bāzē, norādot lietotāja lomu
2.1	Lietotājs var ievadīt jaunu objektu klasifikatoru vai mainīt esošo, laukā „TYPE” izvēloties attiecīgi „objects”.
2.2	Lietotājs var ievadīt jaunu telpu klasifikatoru vai mainīt esošo laukā „TYPE” izvēloties attiecīgi „rooms”
2.3	Lietotājs var meklēt klasifikatoru pēc tā tipa, nosaukuma
3.1	Lietotājs var reģistrēt jaunu darbinieku, norādot viņa personas kodu un pilnu vārdu
3.2.	Lietotājs var mainīt darbinieka vārdu un uzvārdu
3.3	Lietotājs var norādīt darbinieka pilnu dzīves adresi un kontakttālruni vai mainīt esošos
3.4	Lietotājs var piešķirt darbiniekam tipu vai mainīt esošo
3.5	Lietotājs var meklēt darbiniekus pēc vārda un uzvārda, dzīves adreses, kontakttālruņa, tipa
4.1	Lietotājs var reģistrēt jaunu piegādātāju, norādot tā reģistrācijas numuru un nosaukumu
4.2	Lietotājs var norādīt piegādātāja adresi un kontakttālruni vai mainīt esošo
4.3	Lietotājs var meklēt piegādātāju pēc reģistrācijas numura, nosaukuma, adreses
5.1	Lietotājs var reģistrēt jaunu ēku, norādot tās pilnu adresi
5.2	Lietotājs var ievadīt ēkas nosaukumu vai mainīt esošo
5.3	Lietotājs var norādīt ēkas kontakttālruni vai mainīt esošo
5.4	Lietotājs var meklēt ēku pēc ēkas adreses, tipa, kontakttālruņa
6.1	Lietotājs var reģistrēt jaunu telpu, izvēloties ēku, kurā tā atrodas.
6.2	Lietotājs var ievadīt telpas nosaukumu vai mainīt esošo

6.3	Lietotājs var ievadīt telpu izmērus
6.4	Lietotājs var piešķirt telpai vienu no telpu tipiem vai mainīt esošo
6.5	Lietotājs var norādīt ēkas stāvu, kurā atrodas telpa
6.6	Lietotājs var sameklēt telpu pēc nosaukuma, ēkas, stāva, izmēriem, tipa
7.1	Lietotājs var reģistrēt jaunu pasūtījumu, izvēloties no piegādātāju firmām vienu, vai ievadot jaunu
7.2	Lietotājs var ievadīt pavadzīmes numuru
7.3	Lietotājs var norādīt pavadzīmes datumu, izvēloties to kalendārā
7.4	Lietotājs var norādīt piegādes datumu, izvēloties to kalendārā
8.1	Lietotājs var pievienot pasūtījumam rindas
8.2	Lietotājs var pievienot pasūtījumā esošas vienības, izvēloties no vienību saraksta, vai ievadot jaunu.
8.3	Lietotājs var norādīt vienādu vienību vērtību
8.4	Lietotājs var norādīt vienādu vienību skaitu
9.1	Lietotājs var ģenerēt jaunus objektus no pasūtījuma
9.2	Lietotājs var piešķirt numurus jauniem objektiem no pasūtījuma
9.3	Lietotājs var norādīt objekta lietošanas sākuma laiku, izvēloties to no kalendāra
9.4	Lietotājs var pievienot aprakstu par objektu
9.5	Lietotājs var norādīt objekta atrašanās vietu (telpu) vai mainīt esošo
9.6	Lietotājs var izvēlēties par objektu atbildīgu darbinieku vai mainīt esošo
9.7	Lietotājs var norādīt, kurai darba vietai pieder objekts, izvēloties vienu no esošajām vai ievadot jaunu
10.1	Lietotājs var meklēt objektus

10.2	Lietotājs var nomainīt objekta atrašanās vietu (telpu)
10.3	Lietotājs var nomainīt par objektu atbildīgu darbinieku
10.4	Lietotājs var mainīt objekta aprakstu
11.1	Objekta vēsturiski dati tiek automātiski saglabāti, lietotājam mainot objekta telpu, darba vietu, aprakstu,
11.2	Lietotājs var apskatīt objekta vēsturi

Tabula 4. Tabula ar lietotāju stāstiem

2.4 Lietotāju akcepttesti

2.4.1 Akcepttesti

Nr.	Lietotāju akcepttesti
1.1	Lietotājs nevar pieslēgties sistēmai ar nepareizo pieteikumvārdu vai parole
2.1	Lietotājs nevar saglabāt objekta tipu, neizvēloties lauka „TYPE” vērtību „objects”
2.1	Lietotājs nevar saglabāt objekta tipu, ja lauka „Item Name” vērtība sakrīt ar jau eksistējošā objekta tipa lauka „Name” vērtību
2.2	Lietotājs nevar saglabāt objekta tipu, neizvēloties lauka „TYPE” vērtību „rooms”
2.2	Lietotājs nevar saglabāt telpas tipu, ja lauka „Item Name” vērtība sakrīt ar jau eksistējošā telpas tipa lauka „Name” vērtību
3.1	Lietotājs nevar saglabāt darbinieku, ja nav ievadīti personas kods un vārds.
3.1	Lietotājs nevar reģistrēt darbinieku ar jau sistēmā esošo personas kodu
3.4	Lietotājs var piešķirt darbiniekam tipu, kas nozīmēs, ka tas darbinieks būs noklusēti atbildīgs par jauniem objektiem.
3.4	Lietotājs var noņemt no darbinieka atbildīgās par jauniem objektiem personas tipu, tikai, ja uzreiz izvēlēsies citu atbildīgu viņa vietā.
4.1	Lietotājs nevar saglabāt piegādātāju, ja nav ievadīti reģistrācijas numurs, adrese un nosaukums.

4.2	
4.1	Lietotājs nevar saglabāt piegādātāju ar jau eksistējošo sistēmā reģistrācijas numuru.
5.1 5.2	Lietotājs nevar saglabāt ēku, ja nav aizpildīti visi adreses lauki un ievadīti stāvu skaits un ēkas nosaukums.
6.1	Lietotājam vispirms jāizvēlas ēka, tikai tad var ievadīt citus datus
6.1	Ja ēka uzreiz netiek pareizi izvēlēta un tiek mainīta, visi līdz tam ievadītiem datiem jāpazūst
6.1	Lietotājs nevar saglabāt telpu ar jau sistēmā esošo nosaukumu izvēlētajā ēkā.
6.4	Lietotājs var izvēlēties vienu no esošajiem telpu tipiem vai ievadīt jaunu tipu, kas tiks saglabāts pie visiem tipiem.
6.4	Lietotājs nevar saglabāt telpu, neievadot tās tipu un nosaukumu
6.5	Lietotājs nevar norādīt stāvu, kura nav izvēlētajā ēkā
7.1	Skat. 6.1,6.2.
7.2	Lietotājs nevar ievadīt jau esošo sistēmā pavadzīmes numuru
7.3	Izvēloties datumu kalendārā, lietotājs var redzēt to pavadzīmes izrakstīšanas datuma laukā
7.4	Izvēloties datumu kalendārā, lietotājs var redzēt to pasūtījuma saņemšanas datuma laukā
8.4	Lietotājs nevar saglabāt pasūtījuma rindas, ja starp tiem ir rindas, kurām lauku „Item Name” un „Cost” vērtības ir vienādas
8.5	Skat. 2.1.
8.6	Lietotājs nevar ievadīt summu, kas nav formātā 99,99
8.6 8.7	Lietotājs var redzēt pasūtījuma kopējo summu laukā „Total” un summu par vienādām vienībām laukā „Item Total”. Kamēr vienību skaits un vērtība nav ievadīti, laukiem

	„Total” un Item Total” jābūt tukšiem.
9.1	Lietotājs nevar ģenerēt objektus no pasūtījuma atkārtoti
9.2	Ievadot numuru, lietotājs var izmantot punktu
9.3	Izvēloties datumu no kalendāra, lietotājs var to redzēt laukā „Start Date”
9.4	Lietotājs var pievienot aprakstu līdz 100 simboliem garu
9.5	Skat. 3.4.
9.6	Lietotājs var redzēt visus objektus darba vietā, izvēloties vienu no esošajām
9.7	Izvēloties darba vietu no jau esošajām, lietotājs var redzēt visas darba vietas un objektus tajās
9.7	Lietotājs var redzēt izvēlētās darba vietas numuru laukā „Work Place”
10.2	Lietotājs var redzēt datus par objekta telpu blokā „Room”
10.3	Lietotājs var redzēt datus par objekta atbildīgu darbinieku blokā „Responsible Person”
11.2	Lietotājs var redzēt datumu, kad tika veiktas izmaiņas un lietotāju, kas ir veicis izmaiņas
11.2	Lietotājs nevar rediģēt vēsturiskus datus.

Tabula 5. Tabula ar akcepttestiem

3 Programmatūras projektējums

3.1 Ievads

Biroja inventāra uzskaites sistēma (BIUS) datu ievades sadaļa ir paredzēta visa uzņēmuma inventāra uzskaitē, datu sistematizācijai un analīzei, lai nodrošinātu inventāra pārvaldību un izsekošanu uzņēmuma ēkās.

BIUS datu ievades sadaļa nodrošina manuālu datu ievadīšanu par uzņēmuma ēkām, telpā, darbiniekiem, inventāra pasūtījumiem un pasūtījumu piegādātājiem.

3.1.1 Dokumenta nolūks

Dokumenta nolūks ir aprakstīt BIUS ievades sadaļas datu bāzes struktūru un algoritmu projektējumu. Dokuments paredzēts programmaprodukta vadības izstrādātājiem kā tehniskās specifikācijas un apraksta palīgmateriāls.

3.1.2 Dokumenta pārskats

1. nodaļa: Ievads

Nodaļā ir īsi izklāstīts šī dokumenta un sistēmas, uz kuru dokuments attiecas, mērķis un būtība.

Nodaļā sniegts šī dokumenta pārskats.

2. nodaļa: Datubāzes projektējums

Nodaļa satur datubāzes projektējuma diagrammas un tabulu aprakstus.

3. nodaļa: Prasību (lietotāju stāstu) realizācijas apraksts

Šī nodaļa ietver vienā teikumā izklāstus uzdevumus, lai realizētu programmatūras prasības aprakstītas lietotāju stāstos un akcepttestos. Katra uzdevuma numurs sakrīt ar lietotāja stāsta/akcepttesta numuru, kuri ir aprakstīti Programmatūras prasību apraksta dokumentā.

4.nodaļa: Algoritmu UML diagrammas.

Nodaļā ir iekļautas algoritmu UML diagrammas, kuras apraksta kāda konkrēta algoritma realizāciju.

3.2 Datubāzes projektējums

3.2.1 Kopējais ER modelis

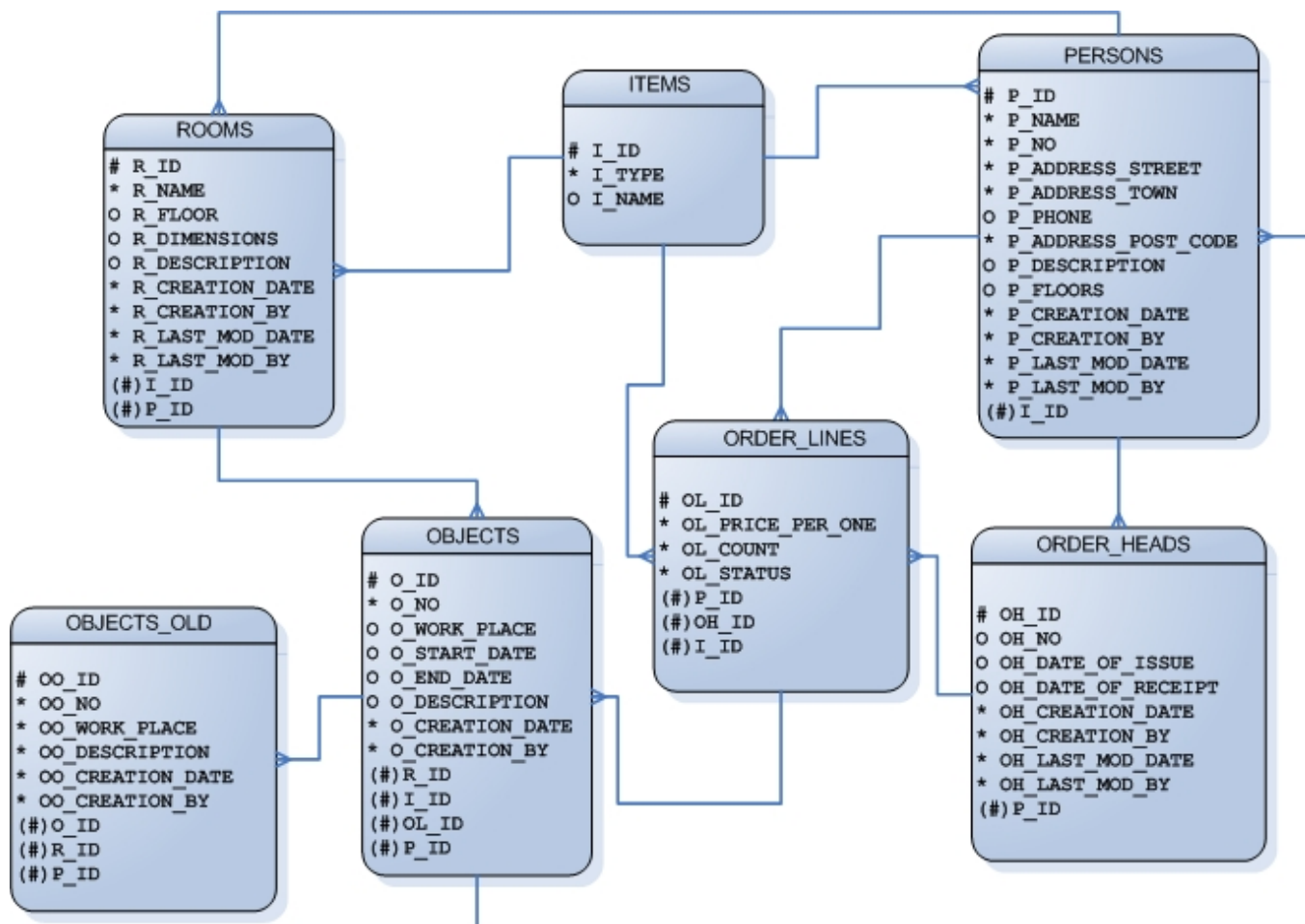


Diagramma 1.ER modelis

3.2.2 Tabulu apraksts

3.2.2.1 Tabula "ITEMS"

Tabula paredzēta citu tabulu datu tipu glabāšanai. Tabulas lauks I_TYPE ir klasifikators, pēc kura var noteikt, kuras tabulas datus apraksta ieraksts. I_TYPE ir vienādi ar to tabulu nosaukumiem, uz kurām attiecas ieraksts.

Primārā atslēga – I_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
I_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'ITEMS_SEQ'
I_TYPE	VARCHAR2(20)		Nē	Tips, kas noteiks, kurai tabulai šis ieraksts ir domāts (tipu nosaukumam jāsakrīt ar tabulas, kurai tas paredzēts, nosaukumu)
I_NAME	VARCHAR2(40)		Nē	Tipam „objects” - inventāra nosaukums, tipam „rooms” - telpas nolūks.

Tabula 6. Tabula „ITEMS”

3.2.2.2 Tabula “ROOMS”

Tabula paredzēta datu par telpām glabāšanai. Lauki R_DESCRIPTION un R_DIMENSIONS var nebūt aizpildīti.

Primārā atslēga – R_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
R_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'ROOMS_SEQ'
R_NAME	VARCHAR2(30)		Nē	Telpas nosaukums
R_FLOOR	NUMBER		Nē	Stāvs, kurā telpa atrodas
R_DESCRIPTION	VARCHAR2(30)		Jā	Svarīgi komentāri
R_DIMENSIONS	VARCHAR2(30)		Jā	Telpas izmēri
R_CREATION_DATE	DATE		Nē	Automātiski (ar trigeru 'ROOMS_NEW' palīdzību) ģenerēts datums pie jaunas telpas izveidošanas)
R_CREATION_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'ROOMS_NEW' palīdzību) ģenerēts lietotāja, kas veic jaunas telpas izveidošanu, vārds)
R_LAST_MOD_DATE	DATE		Nē	Automātiski (ar trigeru 'ROOMS_MOD' palīdzību) ģenerēts datums, veicot jebkādas izmaiņas kādā no kolonām.
R_LAST_MOD_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'ROOMS_MOD' palīdzību) ģenerēts lietotāja, kas veic jebkādas izmaiņas kādā no kolonām, vārds.

Tabula 7. Tabula „ROOMS”

3.2.2.3 Tabula "PERSONS"

Tabula paredzēta datu par darbinieku, piegādātāju un ēku glabāšanai. Uz ko attiecas dati ierakstā var noteikt pēc ārējās atslēgas I_ID, kas satur klasifikatoru.

Primārā atslēga – P_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
P_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'PERSONS_SEQ'
P_NAME	VARCHAR2(30)		Jā	Darbinieka vārds vai ēkas nosaukums, vai piegādātāja firmas nosaukums
P_ADDRESS_STREET	VARCHAR2(20)		Jā	Darbinieka, ēkas vai piegādātāja adreses iela
P_POST_CODE	VARCHAR2(20)		Jā	Darbinieka, ēkas vai piegādātāja pasta indekss
P_PHONE_NO	VARCHAR2(20)		Jā	Darbinieka, ēkas vai piegādātāja kontaktārunis
P_FLOORS	NUMBER		Jā	Stāvu skaits ēkā, citiem tukšs
P_DESCRIPTION	VARCHAR2(50)		Jā	Svarīgs komentārs
I_ID	NUMBER	FK1	Nē	Ārējā atslēga uz tabulas 'ITEMS' (I_ID)
P_ADDRESS_TOWN	VARCHAR2(20)		Jā	Darbinieka, ēkas vai piegādātāja adreses pilsēta

P_CREATION_DATE	DATE		Nē	Automātiski (ar trigeru 'PERSONS_NEW palīdzību ģenerēts datums pie jaunas telpas izveidošanas)
P_CREATION_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'PERSONS_NEW palīdzību ģenerēts lietotājs, kas veic jaunas telpas izveidošanu)
P_LAST_MOD_DATE	DATE		Nē	Automātiski (ar trigeru 'PERSONS_MOD' palīdzību) ģenerēts datums, veicot jebkādas izmaiņas kādā no kolonām.
P_LAST_MOD_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'PERSONS_MOD' palīdzību) ģenerēts lietotāja, kas veic jebkādas izmaiņas kādā no kolonām, vārds.

Tabula 8. Tabula „PERSONS”

3.2.2.4 Tabula “ORDER_HEADS”

Tabula paredzēta datu par pasūtījumiem glabāšanai.

Primārā atslēga – OH_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
OH_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'ORDER_HEADS_SEQ'
OH_NO	VARCHAR2(30)		Nē	Pasūtījuma pavadzīmes numurs

OH_DATE_OF_ISSUE	DATE		Jā	Pasūtījuma pavadzīmes datums
OH_DATE_OF_RECEIPT	DATE		Jā	Pasūtījuma piegādes datums
OH_CREATION_DATE	DATE		Nē	Automātiski (ar trigeru 'ORDER_HEADS_NEW' palīdzību ģenerēts datums pie jaunas telpas izveidošanas)
OH_CREATION_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'ORDER_HEADS_NEW' palīdzību ģenerēts lietotāja, kas veic jaunas telpas izveidošanu, vārds)
OH_LAST_MOD_DATE	DATE		Nē	Automātiski (ar trigeru 'ORDER_HEADS_MOD' palīdzību) ģenerēts datums, veicot jebkādas izmaiņas kādā no kolonām.
OH_LAST_MOD_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'ORDER_HEADS_MOD' palīdzību) ģenerēts lietotāja, kas veic jebkādas izmaiņas kādā no kolonām, vārds.
P_ID	NUMBER	FK1	Nē	Ārējā atslēga uz tabulas 'PERSONS' (P_ID)

Tabula 9. Tabula „ORDER_HEADS”

3.2.2.5 Tabula “ORDER_LINES”

Tabula paredzēta datu par pasūtījumiem glabāšanai.

Primārā atslēga – OL_ID.

Lauks	Tips	Atslēg a	Tukšs	Komentārs
OL_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'ORDER_LINES_SEQ'
OL_PRICE_PER_ONE	NUMBER		Nē	Vienību vērtība
OL_COUNT	NUMBER		Nē	Vienību skaits
OL_STATUS	VARCHAR2(5)		Nē	Rindas statuss
OH_ID	NUMBER		Nē	Ārējā atslēga uz tabulas 'ORDER_HEADS (OH_ID)
I_ID	NUMBER		Nē	Ārējā atslēga uz tabulas 'ITEMS' (I_ID)

Tabula 10. Tabula „ORDER_LINES”

3.2.2.6 Tabula “OBJECTS”

Tabula paredzēta datu par inventāru glabāšanai.

Primārā atslēga – O_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
O_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'OBJECTS_SEQ'
O_NO	VARCHAR2(30)		Jā	Manuāli piešķirts inventāra numurs
O_START_DATE	DATE		Nē	Inventāra ekspluatācijas sākuma datums
O_END_DATE	DATE		Jā	Inventāra ekspluatācijas beigu datums

O_DESCRIPTION	VARCHAR2(50)		Jā	Komentāri
O_CREATION_DATE	DATE		Nē	Automātiski (ar trigeru 'OBJECTS_NEW' palīdzību ģenerēts datums pie jaunas telpas izveidošanas)
O_CREATION_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'OBJECTS_NEW' palīdzību ģenerēts lietotāja, kas veic jaunas telpas izveidošanu, vārds)
I_ID	NUMBER	FK1	Nē	Ārējā atslēga uz tabulas 'ITEMS' (I_ID)
R_ID	NUMBER	FK2	Jā	Ārējā atslēga uz tabulas 'ROOMS' (R_ID)
OL_ID	NUMBER	FK3	Nē	Ārējā atslēga uz tabulas 'ORDER_LINES' (OL_ID)
P_ID	NUMBER	FK4	Nē	Ārējā atslēga uz tabulas 'PERSONS' (P_ID)

Tabula 11. Tabula „OBJECTS”

3.2.2.7 Tabula “OBJECTS_OLD”

Tabula paredzēta vēsturisku datu par inventāru glabāšanai.

Primārā atslēga – OO_ID.

Lauks	Tips	Atslēga	Tukšs	Komentārs
OO_ID	NUMBER	PK	Nē	Automātiski ģenerēts no secības (sequence) 'OBJECTS_OLD_SEQ'
OO_NO	VARCHAR2(30)		Jā	Manuāli piešķirts inventāra

				numurs
OO_DESCRIPTION	VARCHAR2(50)		Jā	Komentāri
OO_CREATION_DATE	DATE		Nē	Automātiski (ar trigeru 'OBJECTS_OLD_NEW' palīdzību ģenerēts datums pie jaunas telpas izveidošanas)
OO_CREATION_BY	VARCHAR2(30)		Nē	Automātiski (ar trigeru 'OBJECTS_OLD_NEW' palīdzību ģenerēts lietotāja, kas veic jaunas telpas izveidošanu, vārds)
I_ID	NUMBER	FK1	Nē	Ārējā atslēga uz tabulas 'ITEMS' (I_ID)
R_ID	NUMBER	FK2	Jā	Ārējā atslēga uz tabulas 'ROOMS' (R_ID)
OL_ID	NUMBER	FK3	Nē	Ārējā atslēga uz tabulas 'ORDER_LINES' (OL_ID)
P_ID	NUMBER	FK4	Nē	Ārējā atslēga uz tabulas 'PERSONS (P_ID)

Tabula 12. Tabula „OBJECTS_OLD”

3.3 Prasību (lietotāju stāstu) realizācijas uzdevumi

Uzdevums	Stāsta Nr.
Izveidot sinonīmus lietotāju piekļuvei visiem datu bāzes objektiem	1.1
Izveidot lietotāju lomu, kas varēs izmantot sistēmu	1.1
Izveidot tabulu klasifikatoru glabāšanai	2.1 2.2

Izveidot formu klasifikatoru attēlošanai un ievadīšanai	2.1 2.2
Izveidot vērtību sarakstu (LOV) klasifikatoru meklēšanai	2.3
Lauku „TYPE” izveidot kā saraksta elementu ar vērtībām „objects” un „rooms”	2.1 2.2
Izveidot algoritmu, kas pievieno objektu klasifikatoru	2.1 2.2
Izveidot vienībtestus objektu klasifikatoru pievienošanas algoritmam	2.1 2.2
Izveidot formu darbinieku attēlošanai un ievadīšanai	3.1
Izveidot vērtību sarakstu (LOV) darbinieku meklēšanai	3.5
Izveidot algoritmu darbiniekam automātiskai klasifikatora pievienošanai	3.1
Izveidot lauku, kur var atzīmēt darbinieku kā atbildīgu par objektiem (klasifikatora manuālā pievienošana)	3.4
Izveidot algoritmu darbiniekam manuālai klasifikatora pievienošanai	3.1
Izveidot vienībtestus darbiniekam automātiskas klasifikatora pievienošanas algoritmam	3.1
Izveidot vienībtestus darbiniekam manuālas klasifikatora pievienošanas algoritmam	3.4
Izveidot darbinieku pievienošanas algoritmu	3.1 3.2
Izveidot vienībtestus darbinieku pievienošanas algoritmam	3.1 3.2
Izveidot formu piegādātāju attēlošanai un ievadīšanai	4.1
Izveidot vērtību sarakstu (LOV) piegādātāju meklēšanai	4.3
Izveidot algoritmu piegādātājam automātiskai klasifikatora pievienošanai	4.1
Izveidot vienībtestus piegādātājam automātiskas klasifikatora pievienošanas algoritmam	4.1

Izveidot piegādātāju pievienošanas algoritmu	4.1 4.2
Izveidot vienībtestus piegādātāju pievienošanas algoritmam	4.1 4.2
Izveidot tabulu ēku (kopīga tabula ar piegādātājiem, darbiniekiem) glabāšanai	5.1
Izveidot formu ēku attēlošanai un ievadīšanai	5.1
Izveidot vērtību sarakstu (LOV) ēku meklēšanai	5.4
Izveidot algoritmu ēkai automātiskai klasifikatora pievienošanai	5.1
Izveidot vienībtestus ēkai automātiskas klasifikatora pievienošanas algoritmam	5.1
Izveidot ēku pievienošanas algoritmu	5.1 5.2 5.3
Izveidot vienībtestus ēku pievienošanas algoritmam	5.1 5.2 5.3
Izveidot tabulu telpu glabāšanai	6.1
Izveidot formu telpu attēlošanai un ievadīšanai	6.1 6.2 6.3
Izveidot vērtību sarakstu (LOV) telpu meklēšanai	6.6
Izveidot pogu ar vērtību sarakstu (LOV) tipa izvēlei	6.4
Izveidot pogu jauna tipa ievadīšanai (ja vajadzīga nav starp esošiem)	6.4
Izveidot lauku „Room Type” telpas tipa attēlošanai.	6.4
Izveidot pogu ar vērtību sarakstu (LOV) ēkas izvēlei	6.5
Izveidot pogu jaunas ēkas ievadīšanai (ja vajadzīgās nav starp esošām)	6.5
Izveidot lauku „Building” ēkas, kur atrodas telpa, attēlošanai	6.5

Izveidot algoritmu, kas automātiski pievieno telpu klasifikatoru	6.1
Izveidot algoritmu telpu pievienošanai	6.1
Izveidot vienībtestus telpu pievienošanas algoritmam	6.1
Izveidot formu pasūtījumu ievadīšanai	7.1
Izveidot tabulu pasūtījumu glabāšanai	7.1
Izveidot pogu piegādātāja izvēlei ar vērtību sarakstu (LOV) piegādātāju meklēšanai	7.1
Izveidot pogu piegādātāja ievadīšanai (ja vajadzīgā nav starp esošiem)	7.1
Izveidot pasūtījuma pievienošanas algoritmu	7.1 7.2
Izveidot vienībtestus pasūtījuma pievienošanas algoritmam	7.1 7.2
Izveidot kalendāru pavadzīmes datuma izvēlēšanai	7.3
Izveidot kalendāru piegādes datuma izvēlēšanai	7.4
Izveidot formu pasūtījuma rindu ievadīšanai	8.1
Izveidot tabulu pasūtījumu rindu glabāšanai	8.1
Izveidot pogu vienības tipa izvēlei ar vērtību sarakstu (LOV) tipa meklēšanai	8.2
Izveidot pogu vienības tipa ievadīšanai (ja vajadzīgā nav starp esošiem)	8.2
Izveidot pasūtījuma rindu pievienošanas algoritmu	8.1
Izveidot vienībtestus pasūtījuma rindu pievienošanas algoritmam	8.1
Izveidot lauku "Item Total" summas par vienādām vienībām aprēķināšanai un attēlošanai	8.3 8.4
Izveidot lauku "Total" pasūtījuma kopējās summas aprēķināšanai un attēlošanai	8.3 8.4
Izveidot formu jaunu objektu attēlošanai un rediģēšanai	9.2
Izveidot algoritmu jaunu objektu ģenerēšanai no pasūtījuma	9.1

Izveidot tabulu objektu glabāšanai	9.1
Izveidot vienībtestus jaunu objektu ģenerēšanas no pasūtījuma algoritmam	9.1
Izveidot algoritmu atbildīgās personas automātiskai piešķiršanai jaunam objektam	9.1
Izveidot vienībtestus atbildīgās personas automātiskai piešķiršanai jaunam objektam algoritmam	9.1
Izveidot algoritmu telpas piešķiršanai jaunam objektam	9.5
Izveidot vienībtestus telpas piešķiršanai jaunam objektam algoritmam	9.5
Izveidot algoritmu jaunu objektu izmaiņu saglabāšanai	9.4 9.7
Izveidot vienībtestus jaunu objektu izmaiņu saglabāšanas algoritmam	9.4 9.7
Izveidot formu objektu attēlošanai	10.1
Izveidot vērtību sarakstu (LOV) objektu meklēšanai	10.1
Izveidot objekta telpas nomainīšanas algoritmu	10.2
Izveidot vienībtestus objekta telpas nomainīšanas algoritmam	10.2
Izveidot algoritmu par objektu atbildīgā darbinieka nomainīšanai	10.3
Izveidot vienībtestus objektu atbildīgā darbinieka nomainīšanas algoritmam	10.3
Izveidot izmaiņu saglabāšanas algoritmu	10.2 10.3
Izveidot vienībtestus izmaiņu saglabāšanas algoritmam	10.2 10.3
Izveidot formu objektu vēstures attēlošanai	11.1
Izveidot tabulu objektu vēstures glabāšanai	11.1
Izveidot pogu, kas atver izvēlētā objekta vēsturi	11.2

Tabula 13. Tabula ar prasību realizācijas uzdevumiem

3.4 Algoritmu UML aktivitāšu diagrammas

3.4.1 Jaunu objektu ģenerēšana no pasūtījuma

Diagramma lietotāja stāstam 9.1. atbilstošam realizācijas algoritmam.

Algoritms jaunu objektu ģenerēšanai no pasūtījuma

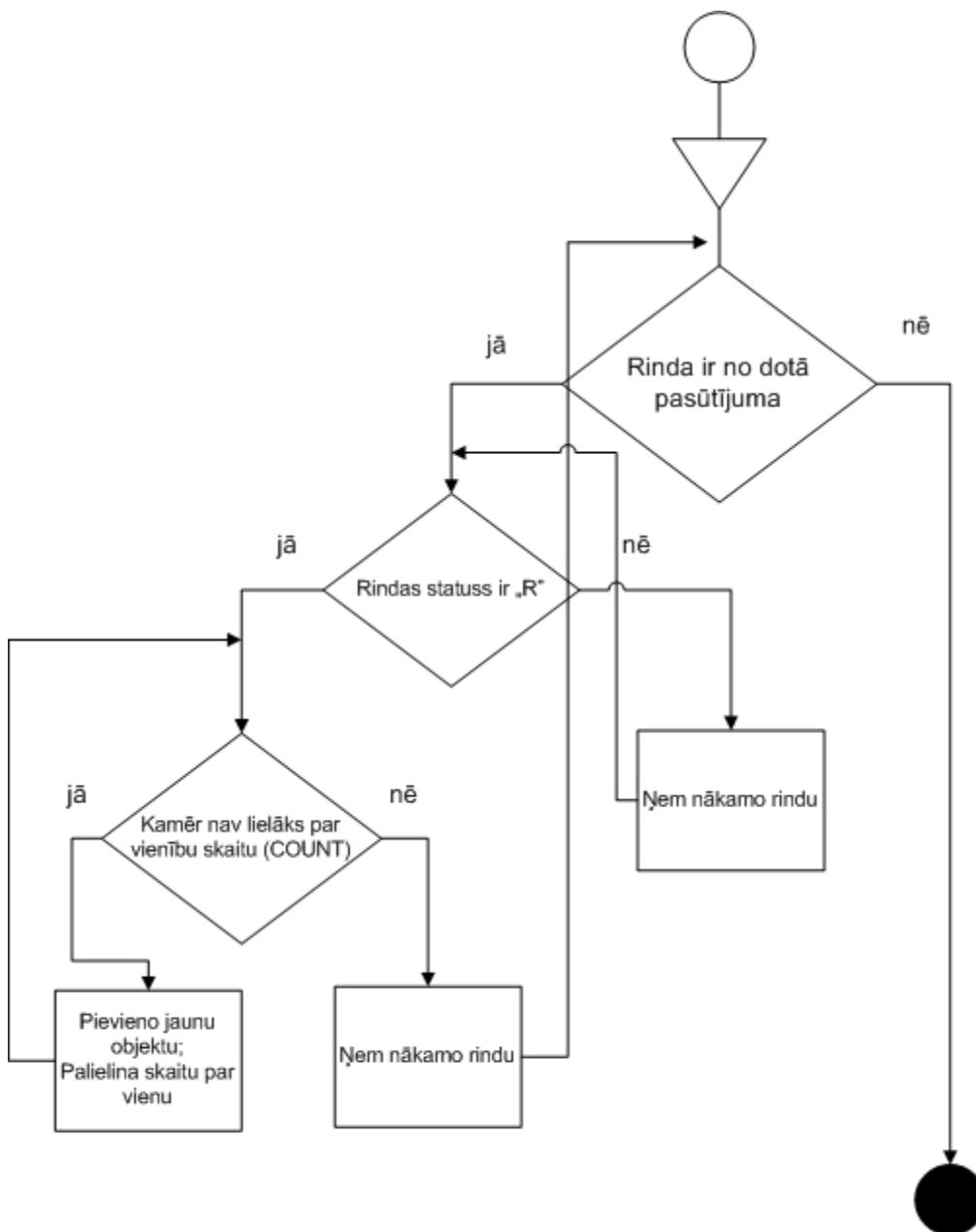


Diagramma 2. Jaunu objektu ģenerēšana no pasūtījuma

3.4.2 Par objektu atbildīgā darbinieka nomainīšana

Diagramma lietotāja stāsta 10.03. atbilstošam realizācijas algoritmam.

Algoritms par objektu atbildīgā darbinieka nomainīšanai

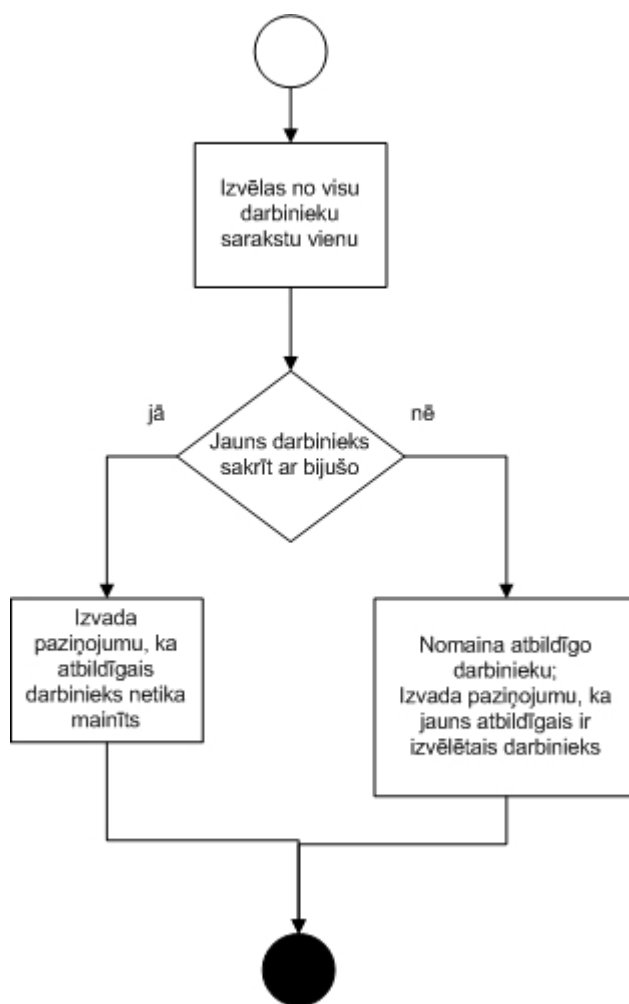


Diagramma 3. Par objektu atbildīgā darbinieka nomainīšana

3.4.3 Objekta vēsturisku datu saglabāšana

Diagramma lietotāja stāsta 11.1.. atbilstošam realizācijas algoritmam.

Algoritms objekta vēsturisku datu saglabāšanai

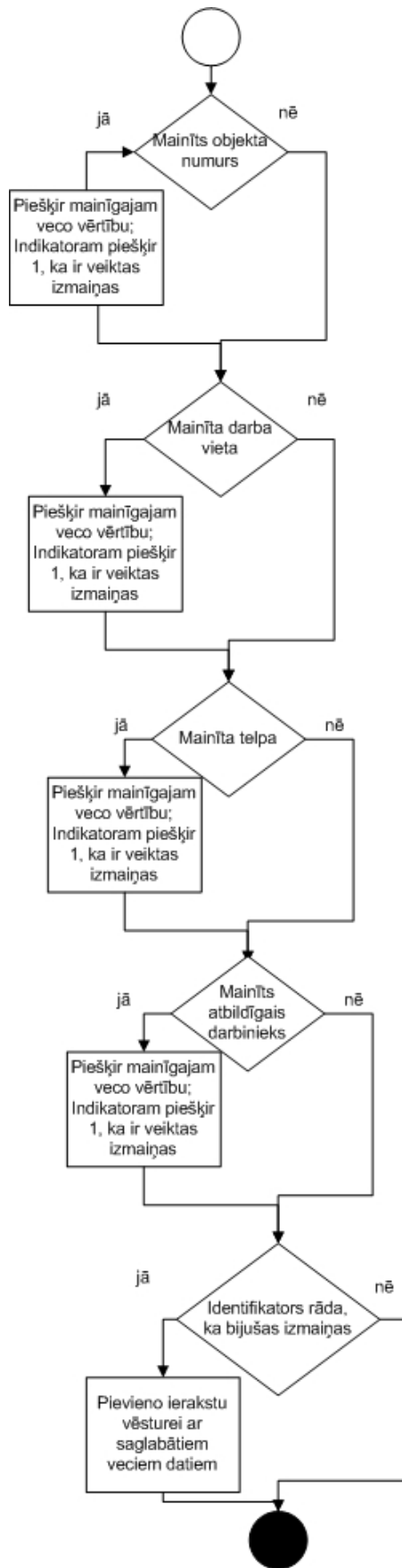


Diagramma 4. Objekta vēsturisku datu saglabāšana

4 Testēšanas dokumentācija

4.1 Nolūks

Testēšanas dokumentācija satur “Biroja inventāra uzskaites sistēmas” (BIUS) vienību testēšanas aprakstu un testu rezultātus.

4.2 Ievads

Vienību testi tika sastādīti, ņemot vērā leitojāju stāstus un akcepttestus. Vienību testēšanas mērķis ir noteikt, vai BIUS funkcionalitāte atbilst izvirzītām prasībām.

Testēšanas procesā funkcionālās prasības tikai testēti manuāli. Testu ievaddati bija izvēlēti tā, lai notestētu vienības, pamatojoties uz Lietotāju stāstiem un akcepttestiem.

4.3 Testu sagatavošana

Testi tika sagatavoti pēc melnās kastes principa, kad tika pieņemts, ka kods nav zināms.

4.4 Testēšanas rezultāti

Vienības testa nosaukums	Izpildes rezultāts (OK/NOK)
„lietotājs var autorizēties sistēmā”	OK
„var atvērt klasifikatoru formu”	OK
„var pievienot objektu klasifikatoru”	OK
„jābūt kļūdas paziņojumam, ja pievieno objektu klasifikatoru ar jau eksistējošo sistēmā objektu klasifikatora lauka „Item Name” vērtību”	OK
„var pievienot telpu klasifikatoru”	OK
„jābūt kļūdas paziņojumam, ja pievieno telpu klasifikatoru ar jau eksistējošo sistēmā telpu klasifikatora lauka „Item Name” vērtību”	OK
„var meklēt klasifikatorus pēc tipa un nosaukuma”	OK
„jābūt kļūdas paziņojumam, ja pievienoklasifikatoru, nenorādot nosaukumu vai neizvēloties klasifikatora tipu”	OK

„var atvērt darbinieku formu”	OK
„var pievienot jaunu darbinieku”	OK
„var atzīmēt darbinieku par atbildīgu par objektiem vai otrādi (noņemt atbildību)”	OK
„noņemot atbildīgās personas statusu no darbinieka, jābūt paziņojumam, ka jāizvēlās cits darbinieks kā atbildīgais un iespējai izvēlēties darbinieku”	OK
„jābūt kļūdas paziņojumam, ja pievieno darbinieku ar jau eksistējošu personas kodu”	OK
„jābūt kļūdas paziņojumam, ja pievieno darbinieku, nenorādot vārdu, adresi, personas kodu”	OK
„var meklēt darbinieku pēc vārda, personas koda, adreses, par objektiem atbildīgās personas statusa”	OK
„var atvērt piegādātāju formu”	OK
„var pievienot jaunu piegādātāju”	OK
„jābūt kļūdas paziņojumam, ja pievieno piegādātāju ar jau eksistējošu reģistrācijas numuru”	OK
„jābūt kļūdas paziņojumam, ja pievieno piegādātāju, nenorādot nosaukumu, adresi, reģistrācijas numuru”	OK
„var meklēt piegādātāju pēc nosaukuma, reģistrācijas numura, adreses”	OK
„var atvērt ēku formu”	OK
„var pievienot jaunu ēku”	OK
„jābūt kļūdas paziņojumam, ja pievieno ēku ar jau eksistējošu nosaukumu”	OK
„jābūt kļūdas paziņojumam, ja pievieno ēku, nenorādot nosaukumu, adresi”	OK
„var meklēt ēku pēc nosaukuma, adreses, stāvu skaita”	OK

„var atvērt telpu formu”	OK
„pievienojot jaunu telpu, var izvēlēties ēku”	OK
„pievienojot jaunu telpu, var ievadīt un izvēlēties jaunu ēku”	OK
„pievienojot jaunu telpu, var izvēlēties telpas tipu”	OK
„pievienojot jaunu telpu, var izvēlēties ēku”	OK
„var pievienot jaunu telpu”	OK
„jābūt kļūdas paziņojumam, ja pievieno telpu ar jau eksistējošu nosaukumu izvēlētajā ēkā”	OK
„jābūt kļūdas paziņojumam, ja pievieno telpu ar stāvu, kas nav izvēlētajā ēkā”	OK
„jābūt kļūdas paziņojumam, ja pievieno telpu, nenorādot nosaukumu, tipu vai ēku”	OK
„var meklēt telpu pēc nosaukuma, ēkas, stāva un tipa (klasifikatora tipa)”	OK
„var atvērt pasūtījumu formu”	OK
„var izvēlēties piegādātāju”	OK
„jābūt kļūdu paziņojumam, ja piegādātājs nav izvēlēts”	OK
„var pievienot jaunu pasūtījumu”	OK
„jābūt kļūdas paziņojumam, ja pievieno pasūtījumu ar jau eksistējošu pavadzīmes numuru”	OK
„jābūt kļūdas paziņojumam, ja pievieno pasūtījumu, nenorādot pavadzīmes numuru, pavadzīmes datumu un piegādes datumu”	OK
„var izvēlēties tipu”	OK
„jābūt kļūdu paziņojumam, ja tips nav izvēlēts”	OK
„var pievienot jaunu pasūtījuma rindu”	OK

„jābūt kļūdas paziņojumam, ja pievieno pasūtījuma rindu, kuras lauku „Item Name” un „Cost” vērtības ir vienādas ar citu pasūtījuma rindu”	OK
„var ģenerēt jaunus objektus”	OK
„var apskatīt uzģenerētus objektus”	OK
„var piešķirt objektiem numurus”	OK
„jābūt kļūdu paziņojumam, ja numurs netiek piešķirts”	OK
„var saglabāt izmaiņas”	OK
„var meklēt objektu pēc objekta numura, ekspluatācijas sākuma datuma, tipa, telpas, ēkas, atbildīgās personas”	OK
„var nomainīt objekta telpu”	OK
„var nomainīt atbildīgu personu”	OK
„var mainīt objekta aprakstu”	OK
„var apskatīt izvēlētā objekta vēsturi”	OK

Tabula 14. Tabula ar testēšanas rezultātiem

5 Projekta organizācija

Programmprodukts „Biroja inventāra uzskaites sistēma” (BIUS) tika izstrādāts pēc Spējās programmatūras izstrādes modeļa.

Spējās metodoloģija ir uz noteikumiem balstīta vienota pieeja programmatūras izstrādei. Tā samazina projekta risku pateicoties iespējai viegli reaģēt uz projekta izmaiņām un absorbēt tās, netraucējot piegādāt izstrādājamo sistēmu paredzētajā laikā. Kas īpaši piemērota izstrādei, kad autors paralēli apgūst jaunas tehnoloģijas. Sadalot visu darbu nelielās distancēs (sprint) un bieži atrādot produktu klientam, projekts tika pasargāts no iespējas ciest pilnīgu neveiksmi, jo visas nepilnības un neatbilstības prasībām tika novērstas uzreiz. Spejas programmatūras izstrādes plānošana tika realizēta ar Scrum metodoloģijas palīdzību.

Sākumā tika izveidots saraksts ar sākumā identificēto sistēmas funkcionalitāti (product backlog). Sistēmas funkcionalitātes saraksts tika veidots no lietotāju stāstiem (*user stories*), kas ir nelieli sistēmas lietošanas scenāriji no lietotāja viedokļa.

Lietotāju stāsti tika sakārtoti prioritāšu secībā, atbilstoši tam, kādā secībā klients vēlas, lai funkcionalitāte tiktu realizēta. No klienta puses tika izvēlēts produkta īpašnieks (*product owner*), kas ir primāri atbildīgs, par lietotāju stāstu sakārtošanu prioritāšu secībā.

Lietotāju stāsti tika sagrupēti pa laidieniem (*releases*) jeb kārtām. Viena laidienā jeb kārtā tika iekļauti lietotāju stāsti, kurus ir plānots kopīgi uzstādīt produkcijā pēc tam, kad tie būs realizēti.

Laidiena jeb kārtas lietotāju stāstu realizācija tika veikta ar sprintu palīdzību, vienam sprintam tika sagrupēti visi uzdevumi, kuri jāizpilda, lai realizētu kādu konkrētu lietotāju stāstu (*sprint backlog*).

Lai izsekotu produkta izstrādes progresu, tika zīmēts “sadeģšanas” grafiks (Burndown Chart). No sākuma tika novilkta ideāla progresā līnija, pēc katras ikdienas Scrum sanāksmes tika zīmēta reāla progresā līnija. Ja tā stipri atšķirtos no ideālās, būtu skaidrs, ka projektā ir radušas problēmas, kuras pēc iespējas ātrāk jānovērš.

Kvalifikācijas darba ietvaros sprinti bija 2 nedēļu gari, lai novērstu pārpratamības iespējas pašā sākumā.

6 Kvalitātes nodrošināšana

Lai nodrošinātu kvalitatīva produkta izstrādi, tika veiktas šādas darbības:

- Regulārās programmatūras prasību pārrunas ar klientu, prasību regulāra pierakstīšana Scrum (*product backlog*) dokumentā.
- Programmatūras manuālu vienības testu veidošana un palaišana paralēli izstrādes procesam, pamatojoties uz lietotāju akcepttestiem;
- Projekta Scrum (*product backlog, sprint backlog*) dokumenta regulāra atjaunošana;
- Programmprodukta kods tika pilnīgi komentēts angļu valodā. Sarežģītākām procedūrām un funkcijām tika veidoti komentāri par to iekšējo darbību.

7 Konfigurācijas pārvaldība

Programmprodukta un dokumentācijas konfigurāciju pārvaldībai tikai izmantota Subversion (SVN) versiju kontroles sistēma, kura ir paredzēta, lai vēsturiski uzturētu un kontrolētu izejas kodus, dokumentācijas un citas lietas. Šīs sistēmas izmantošana bija īpaši ērta, jo SVN ļauj jebkurā brīdī atgriezties pie senāka ieraksta vai faila, pēc vajdzības atjaunot to no kādas citas versijas, kas ir svarīgi, izstrādājot Oracle formas.

Uz izstrādes datora tika uzinstalēts klients (TortoiseSVN Version 1.6.5.), kas pieslēdzās datu repozitorijam uz kalpdatora, izmantojot http protokolu.

jaunas formas, vai veicot izmaiņas esošajās, kā arī ievietojot procedūru un funkciju kodu, tika pievienoti pēc iespējas detalizētāki komentāri, lai nodrošinātu efektīvāku izmaiņu pārvaldību. Tāpat tika uzsverts, ka katrai produkta versijai, kas tiek saglabāta, jābūt strādājošai.

8 Darbietilpības novērtējums

Veicot programmatūras izstrādi pēc Spējās metodoloģijas, tika izmantota arī attiecīga darbietilpības novērtēšanas prakse.

1. Tika izmantota sarežģītības skalas punkti 1,2,3,4,5,8, kuru apraksts ir tabulā

Sarežģītības punkts	Apraksts
1	Ļoti vienkāršs
2	Vienkāršs
3	Gandrīz vidēji
4	Vidēji
5	Gandrīz sarežģīti
8	Sarežģīti

Tabula 15. Sarežģītības punktu skala

2. Rēķinot darbietilpību, tika izmantotas šādas vērtības:

- 1 cilvēkdiena = 8 cilvēkstundas;
- 1 cilvēk mēnesis = 20 cilvēkdienas = 160 cilvēkstundas;
- 1 kalendāra mēnesis = 30 cilvēkdienas = 360 cilvēkstundas = 1 cilvēk mēnesis/2,25

3. Tika noteikts viena sprinta garums – 2 nedēļas, t.i. 80 ideālas cilvēkstundas, ņemot vērā, ka risinājuma izstrāde tiek iesaistīts tikai 1 cilvēks;

4. Tika izrēķināts sprinta ātrums, t.i. cik sarežģītības punkti var būt nerealizēti vienā sprinta garumā, pēc formulas:

Sprintu ātrums = 80 * Lietotāja stāsta novērtējums sarežģītības punktos /

Kopējais uzdevumu novērtējums cilvēkstundas

5. Tika izrēķināta kopējā darbietilpība ideālās cilvēkstundās pēc formulas:

Sprintu skaits = Darba kopējais apjoms sarežģītības punktos / sprinta ātrums

Ideālas cilvēkstundas = Sprintu skaits * 80

6. Tika noteikti darba realizācijas laika ietekmējošie faktori, un atkarība no to ietekmes pakāpes tika aprēķināta galēja darbietilpība:

Ietekmes pakāpe	Darbietilpība
Zema	Darbietilpība = darbietilpība (ideālās cilvēkstundas) * (1+0.3), t.i. +30%
Vidēja	Darbietilpība = darbietilpība (ideālās cilvēkstundas) * (1+0.5), t.i. +50%
Augsta	Darbietilpība = darbietilpība (ideālās cilvēkstundas) * (1+0.7), t.i. +70%

Tabula 16. Ietekmes pakāpes

Kvalifikācijas darba izstrādē ietekmes pakāpe bija zema.

7. Tika noteikti sarežģītības punkti visiem augstākā līmeņa prasībām

1. Risinājumam jānodrošina iespēja lietotājiem autentificēties un autorizēties sistēmā				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
7	7	ļoti vienkāršs	0	2

Tabula 17. 1.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

2. Risinājumam jānodrošina iespēja uzturēt objektu , telpu, ēku, piegādātāju klasifikatorus				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
1	5	vienkāršs	4	2

Tabula 18. 2.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

3. Risinājumam jānodrošina iespēja manuāli uzturēt darbiniekus				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits

1	4	vienkāršs	12	5
---	---	-----------	----	---

Tabula 19. 3.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

4. Risinājumam jānodrošina iespēja manuāli uzturēt piegādātāju firmas

Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
1	2	vienkāršs	12	5

Tabula 20. 4.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

5. Risinājumam jānodrošina iespēja manuāli uzturēt ēkas

Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
1	2	vienkāršs	12	5

Tabula 21. 5.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

6. Risinājumam jānodrošina iespēja manuāli uzturēt telpas

Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
1	2	vienkāršs	12	5

Tabula 22. 6.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

7. Risinājumam jānodrošina iespēja manuāli uzturēt pieņemtos pasūtījumus

Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
3	10	vidēji	20	25

Tabula 23. 7.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

8. Risinājumam jānodrošina iespēja manuāli uzturēt pieņemto pasūtījumu rindas				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
2	6	vidēji	8	8

Tabula 24. 8.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

9. Risinājumam jānodrošina iespēja automatizēti uzturēt objektus				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
3	7	vidēji	11	7

Tabula 25. 9.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

10. Risinājumam jānodrošina iespēja manuāli uzturēt objektus				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
6	20	vidēji	22	35

Tabula 26. 10.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

11. Risinājumam jānodrošina iespēja uzturēt objektu vēsturi				
Tabulu skaits	Funkciju/procedūru/trigeru skaits	Funkciju/procedūru/trigeru kopējā sarežģītība	Lauku skaits ekrānformā(s)	Kopējais sarežģītības punktu skaits
2	5	vienkāršs	6	8

Tabula 27. 11.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits

8. Sprinta ātruma un darbietilpības aprēķins

Tika paņemta 2.augstākā līmeņa prasība „Risinājumam jānodrošina iespēja uzturēt objektu , telpu, ēku, piegādātāju klasifikatorus” un visi lietotāju stāsti sadalīti smaklos uzdevumos, kuriem noteiktas cilvēkstundas.

Uzdevums	Lietotāju stāsta Nr.	Cilvēkstundas
Izveidot tabulu klasifikatoru glabāšanai	2.1 2.2	1
Izveidot formu klasifikatoru attēlošanai un ievadīšanai	2.1 2.2	4½
Izveidot vērtību sarakstu (LOV) klasifikatoru meklēšanai	2.3	1/4
Lauku „TYPE” izveidot kā saraksta elementu ar vērtībām „objects” un „rooms”	2.1 2.2	1/4
Izveidot algoritmu, kas pievieno objektu klasifikatoru	2.1 2.2	3
Izveidot vienībtestus objektu klasifikatoru pievienošanas algoritmam	2.1 2.2	3

Tabula 28. 2.augstākā līmeņa prasības realizācijas cilvēkstundu aprēķins

Novērtējums ideālās cilvēkstundās: 12

Darba apjoms sarežģītības punktos: 2

Sprinta ātrums = $80 * 2 / 12 = 13,33$ sarežģītības punkti var būt realizēti viena sprinta garumā

Kopējā darbietilpība = $107 / 13,33 * 80 = 642,16$ ideālās cilvēkstundas

Kopējā darbietilpība = $642,16 * (1+0,3) = 834,808$ cilvēkstundas = **104,35** cilvēkdienas = **5,22** cilvēk

mēneši = **2,32** kalendāru mēneši.

Faktiskā darbietilpība

Faktiskā darbietilpība tika noteikta, pamatojoties uz Scrum dokumenta, kur tiek fiksēts:

- Plānotais laiks – cilvēkstundu skaits, kas bija paredzēts katra sprinta uzdevumiem, kas jau tika uztaisīti vai šobrīd tiek veikti , t.i. 80 cilvēkstundas;
- Faktiskais laiks – cilvēkstundu skaits, kas tika reāli patērēts sprinta laika uzdevumu veikšanai.

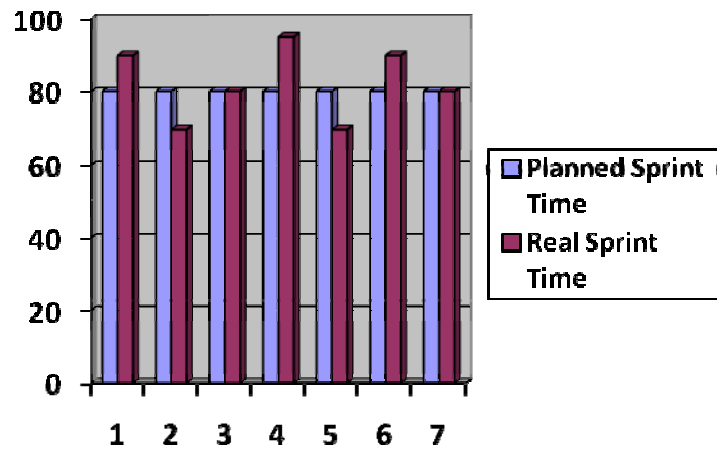


Diagramma 5. Faktiskais sprint garums

Pamatojoties uz diagrammas faktiskā darbietilpība = 535 cilvēkstundas = 3,34 cilvēk mēneši.

9 Rezultāti

Darba izstrādes gaitā tika izveidota pilnīgi funkcionējoša Biroja inventāra uzskaites sistēma.

Darba gaitā autors iepazinās ar Spējās programmatūras izstrādes posmiem, darbietilpības novērtēšanas metodi, izstrādājot programmproduktu ar spējās modeli un apguva Oracle tehnoloģijas un darbu ar Oracle Form Builder sistēmas izstrādei vajadzīgā līmenī, kā arī PL/SQL programmēšanas valodas līdzekļus.

10 Secinājumi

Pēc darba paveikšanas autors secināja, ka spējās programmatūras izstrādes modelis bija izvēlēts pareizi un sekmēja projekta realizācijai, jo tika novērsts risks, ka projekts cietīs pilnīgu neveiksmi.

Autors secināja, ka nākotnē testēšanai būtu noderīgi apgūt kādu ietvaru (framework), jo automatizēti testi pēc pieredzes tiek izpildīti biežāk.

11 Izmantotā literatūra un avoti

- 1) PL/SQL User's Guide and Reference 10g Release 2 (10.2)[tiešsaite] [atsauce 30.04.2010.]. Pieejams: http://downloaduk.oracle.com/docs/cd/B19306_01/appdev.102/b14261.pdf.
- 2) SQL and PL/SQL [tiešsaite] [atsauce 30.04.2010.]. Pieejams: [http://ftp.ku.ac.th/pub/manual/Oracle%209i%20SQL%20Reference%20\(a90125\).pdf](http://ftp.ku.ac.th/pub/manual/Oracle%209i%20SQL%20Reference%20(a90125).pdf).
- 3) Concepts 10g Release 2 (10.2) [tiešsaite] [atsauce 30.04.2010.]. Pieejams: http://download-west.oracle.com/docs/cd/B19306_01/server.102/b14220.pdf.
- 4) About Static Data Dictionary Views [tiešsaite] [atsauce 05.04.2010.]. Pieejams: http://download.oracle.com/docs/cd/B19306_01/server.102/b14237/statviews_1001.htm.
- 5) Бсѣ оѢ Oracle [tiešsaite] [atsauce 05.05.2010.]. Pieejams: <http://baks.gaz.ru/>.
- 6) **These course notes are based on: Oracle Designer/2000 Release 1.3, System Modeling and Tools.** Copyright © Oracle Corporationm 1992, 1995.

12 Programmatūras koda fragmenti

12.1 Ievads

Šajā nodaļā ir iekļauti programmaprodukta koda fragmenti, kas sadalīti grupās:

- Tabulu skripti
- Secības (sequences)
- Trigeri
- Procedūras

12.2 Tabulu skripti

```
--Table "ITEMS"
CREATE TABLE "ITEMS"
( "I_ID" NUMBER NOT NULL ENABLE,
  "I_TYPE" VARCHAR2(20),
  "I_NAME" VARCHAR2(40),
  CONSTRAINT "ITEMS_PK" PRIMARY KEY ("I_ID") ENABLE
)

--Table "PERSONS"
CREATE TABLE "PERSONS"
( "P_ID" NUMBER,
  "P_NAME" VARCHAR2(30) CONSTRAINT "P_NAME_NN" NOT NULL ENABLE,
  "P_NO" VARCHAR2(30),
  "P_ADDRESS_STREET" VARCHAR2(20),
  "P_ADDRESS_TOWN" VARCHAR2(20),
  "P_POST_CODE" VARCHAR2(20),
  "P_PHONE_NO" VARCHAR2(20),
  "P_DESCRIPTION" VARCHAR2(50),
  "P_CREATION_DATE" VARCHAR2(30),
  "P_CREATION_BY" VARCHAR2(30),
  "P_LAST_MOD_DATE" VARCHAR2(30),
  "P_LAST_MOD_BY" VARCHAR2(30),
  "I_ID" NUMBER,
  FOREIGN KEY ("I_ID") REFERENCES "ITEMS" ("I_ID"),
  CONSTRAINT "P_ID_PK" PRIMARY KEY ("P_ID") ENABLE
)
```

12.3 Secības (sequences)

```
--Sequence for table "ITEMS"
CREATE SEQUENCE ITEMS_SEQ
  MINVALUE 1
  START WITH 1
  INCREMENT BY 1
  CACHE 20;
```

12.4 Trigeri

--Triger that saves creation date and user when new record is entered in table "ORDER_HEADS"

```
CREATE OR REPLACE TRIGGER ORDER_HEADS_NEW
BEFORE insert
ON ORDER_HEADS
FOR EACH ROW
```

```
DECLARE
```

```
CREATION_BY      varchar2(40);
CREATION_DATE    date;
```

```
BEGIN
```

```
SELECT
    user,sysdate
INTO
    CREATION_BY, CREATION_DATE
FROM dual ;
```

```
:NEW.OH_CREATION_BY := CREATION_BY ;
:NEW.OH_CREATION_DATE := CREATION_DATE ;
```

```
END;
```

```
/
```

--Triger which saves date and user when table "ORDER_HEADS" is updated or a new recorded is inserted

```
CREATE OR REPLACE TRIGGER ORDER_HEADS_MOD
BEFORE insert or update
ON ORDER_HEADS
FOR EACH ROW
```

```
DECLARE
```

```
LAST_MOD_BY      varchar2(40);
LAST_MOD_DATE    date;
```

```
BEGIN
```

```
SELECT
    user,sysdate
INTO
    LAST_MOD_BY, LAST_MOD_DATE
FROM dual ;
```

```
:NEW.OH_LAST_MOD_BY := LAST_MOD_BY ;
:NEW.OH_LAST_MOD_DATE := LAST_MOD_DATE ;
```

```
END;
```

```
/
```

*--Triger which executes when specific columns of table "OBJECTS" are updated
--and saves history data in table "OBJECTS_OLD"*

```
CREATE OR REPLACE TRIGGER MAKE_OLD
```

```

BEFORE update
ON OBJECTS
FOR EACH ROW

declare
    --"dummy" indicates if changes were made
    dummy number := 0;
    obj varchar2(30) :=NULL;
    WP varchar2(30) :=null;
    room number :=null;
    person number :=null;

BEGIN
    --checks if any changes were made and tells indicator "dummy" 1 if were
    if :old.o_no = :new.o_no then
        obj := NULL;
    else
        obj := :old.o_no;
        dummy := 1;
    end if;

    if :old.o_Work_place = :new.O_Work_place then
        WP := NULL;
    else
        WP := :old.o_Work_place;
        dummy := 1;
    end if;

    if :old.R_ID = :new.R_ID then
        room := NULL;
    else
        room := :old.R_ID;
        dummy := 1;
    end if;

    if :old.P_ID = :new.P_ID then
        person := NULL;
    else
        person := :old.P_ID;
        dummy := 1;
    end if;
    --saves history data if changes were made
    if dummy = 1 then

        INSERT INTO OBJECTS_OLD (OO_ID, OO_NO, OO_WORK_PLACE,
            OO_CREATION_DATE, OO_CREATION_BY, O_ID, R_ID, P_ID)
        values (OBJECTS_OLD_SEQ.nextval, obj, wp, sysdate,
            user, :old.O_id, room, person);
    end if;

END;
/

```

12.5 Procedūras

--Procedure which changes object's responsible person

```
create or replace procedure change_person (o in objects.o_id%TYPE, p in
persons.p_id%TYPE) is
```

```
begin
```

```
    update objects set p_id = p where o_id = o;
    commit;
```

```
end;
```

```
/
```

*--Procedure which generates new objects from order lines of given order if
line status is 'R' and changes status to 'F'*

```
CREATE OR REPLACE PROCEDURE generate_obj (p IN order_lines%ROWTYPE) IS
BEGIN
```

```
    FOR i IN 1..p.ol_count
```

```
    LOOP
```

```
        IF p.ol_status = 'R' then
```

```
            INSERT INTO objects (o_id, i_id, ol_id)
```

```
            VALUES (objects_seq.nextval,p.i_id, p.ol_id);
```

```
            UPDATE order_lines SET ol_status = 'F'
```

```
            WHERE order_lines.ol_status = p.ol_status;
```

```
        END IF;
```

```
    END LOOP;
```

```
END;
```

```
/
```

13 Diagrammu saraksts

Diagramma 1.ER modelis	6
Diagramma 2.Jaunu objektu ģenerēšana no pasūtījuma.....	29
Diagramma 3. Par objektu atbildīgā darbinieka nomainīšana.....	30
Diagramma 4.Objekta vēsturisku datu saglabāšana	31
Diagramma 5.Faktiskais sprint garums	44

14 Tabulu saraksts

Tabula 1. Tabula ar izmantotiem apzīmējumiem	7
Tabula 2. Tabula ar izmantotiem saīsinājumiem.....	7
Tabula 3. Tabula ar augstākā līmeņa prasībām	9
Tabula 4. Tabula ar lietotāju stāstiem.....	12
Tabula 5. Tabula ar akcepttestiem.....	14
Tabula 6. Tabula „ITEMS”	17
Tabula 7. Tabula „ROOMS”	18
Tabula 8. Tabula „PERSONS”	20
Tabula 9. Tabula „ORDER_HEADS”	21
Tabula 10. Tabula „ORDER_LINES”	22
Tabula 11. Tabula „OBJECTS”	23
Tabula 12. Tabula „OBJECTS_OLD”	24
Tabula 13. Tabula ar prasību realizācijas uzdevumiem	28
Tabula 14. Tabula ar testēšanas rezultātiem.....	35
Tabula 15. Sarežģītības punktu skala	39
Tabula 16. Ietekmes pakāpes.....	40
Tabula 17. 1.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	40
Tabula 18. 2.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	40
Tabula 19. 3.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	41
Tabula 20. 4.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	41
Tabula 21. 5.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	41
Tabula 22. 6.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	41
Tabula 23. 7.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	41
Tabula 24. 8.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	42
Tabula 25. 9.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits.....	42

Tabula 26. 10.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits	42
Tabula 27. 11.augstākā līmeņa prasības realizācijas sarežģītības punktu skaits	42
Tabula 28. 2.augstākā līmeņa prasības realizācijas cilvēkstundu aprēķins	43

Kvalifikācijas darbs „**Biroja inventāra uzskaites sistēma**” 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: **Jelena Fjodorova** _____ **31.05.2010.**

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: **M.dat. Aivars Niedrītis** _____ **31.05.2010.**

Recenzents: **M.dat. Pēteris Krastiņš**

Darbs iesniegts 31.05.2010.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Dainis Dosbergs* _____

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

09.06.2010. prot. Nr. _____, vērtējums _____ (_____)

Komisijas sekretārs(-e): _____