

LATVIJAS UNIVERSITĀTE

Datorikas fakultāte

**PROMBŪTNES UZSKAITES MODUĻA IZSTRĀDE
PERSONĀLVADĪBAS SISTĒMĀ
KVALIFIKĀCIJAS DARBS**

Autors: Rihards Mozaļevskis

Studenta apliecības. Nr. rm11055

Vadītājs: M. Dat. Matīss Rikters,

Tartu Universitāte, viespētnieks

Rīga, 2017

ANOTĀCIJA

”PROMBŪTNES UZSKAITES MODUĻA IZSTRĀDE PERSONĀLVADĪBAS SISTĒMĀ” ir kvalifikācijas darbs, kurā tikai izstrādāts uzkrātā atvaļinājuma uzskaites modulis tīmekļa mākoņrisinājumam “Tildes Jumis - Personālvadība”. Izstrādātais modulis nodrošina iespēju lietotājiem noteikt, pēc kādiem kritērijiem aprēķināt uzkrāto atvaļinājumu, kā arī to aprēķināt, ņemot vērā citos sistēmas moduļos reģistrētos notikumus – darba attiecību sākšanas datumu, jau izmantotos atvaļinājumus, informāciju par bērniem u.c.

Modulis tik izstrādāts pēc testu virzītās izstrādes principa. Sistēma ir izstrādāta izmantojot *Microsoft .NET Core* ietvaru, *Angular* ietvaru un citas jaunākās tīmekļa tehnoloģijas.

Atslēgvārdi: uzņēmuma vadības sistēma, personālvadība, tīmekļa programmatūra.

ABSTRACT

“DEVELOPMENT OF ABSENCE ACCOUNTING MODULE OF HUMAN RESOURCE MANAGEMENT SYSTEM” is a thesis in which the author developed a module for calculating the accumulated vacation time of employees. The module is a part of a cloud-based web application “Tildes Jumis - Personālvadība”. The developed module lets a user define how the available vacation days will be calculated and calculates them, considering information that is registered in other system modules, like date when employee has start working, information about children and other information.

The module is developed following the test-driven development approach. The system is developed using *Microsoft .NET Core* framework, *Angular* framework and other latest web technologies.

Keywords: business management system, human resources, web application.

S A T U R S

ANOTĀCIJA	2
ABSTRACT	3
IEVADS.....	6
APZĪMĒJUMU SARAKSTS	7
1 PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA.....	8
1.1 IEVADS	8
1.2 VISPĀRĒJAIS APRAKSTS.....	8
1.3 KONKRĒTĀS PRASĪBAS	10
2 PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS.....	16
2.1 IEVADS	16
2.2 DEKOMPOZĪCIJAS APRAKSTS	17
2.3 DATUBĀZES PROJEKTĒJUMS	23
2.4 EKRĀNFORMU PROJEKTĒJUMS.....	26
2.5 NEFUNKCIONĀLO PRASĪBU REALIZĀCIJA	28
3 TESTĒŠANAS DOKUMENTĀCIJA	29
3.1 IEVADS	29
3.2 TESTĒŠANAS APRAKSTS.....	29
3.3 TESTĒŠANAS ŽURNĀLS	29
4 PROJEKTA ORGANIZĀCIJA	32
5 KONFIGURĀCIJAS PĀRVALDĪBA	33
6 KVALITĀTES NODROŠINĀŠANA	34

7	DARBIETILPĪBAS NOVĒRTĒŠANA.....	35
8	SECINĀJUMI	37
9	IZMANTOTĀ LITERATŪRA.....	38
10	PIELIKUMI	39
	KODA FRAGMENTI.....	39
	10.1.....	39
	10.2 DATUBĀZES REALIZĀCIJAS ER MODELIS	41

IEVADS

Kvalifikācijas darbā izstrādāts prombūtnes uzskaites modulis, precīzāk, uzkrātā atvaļinājuma uzskaites modulis, kas ir daļa no topošās personāla vadības sistēmas Tildes Jumis – Personālvadībā (TJ-PV). Kvalifikācijas darba tēma tika izvēlēta, pamatojoties uz darba autora darba vietas plānotajiem darbiem, kā arī pamatojoties uz studiju laikā iegūtajām zināšanām un apgūtajām tehnoloģijām.

Darba tēma ir aktuāla, jo aizvien vairāk uzņēmumu meklē veidus, kā ērti un kvalitatīvi veikt sava uzņēmuma grāmatvedības un darbinieku pārvaldīšanu. Personāla vadības sistēmas ir neatņemama mūsdienīgu uzņēmumu sastāvdaļa un prombūtnes uzskaitē, kas ir konkrēti šī darba tēma, ir svarīga personāla vadības sistēmas sastāvdaļa.

TJ-PV ir mūsdienīgs mākoņrisinājums, kas pieejams no interneta pārlūkprogrammas. Darbs tika strukturēts septiņās nodaļās. Pirmajā nodaļā ir aprakstīta izstrādājamā moduļa programmatūras prasību specifikācija, otrajā nodaļā ir aprakstīts programmatūras projektējuma apraksts, trešajā – testēšanas dokumentācija. Ceturtnā nodaļa satur projekta organizācijas aprakstu, piektajā ir aprakstīts tas, kā projektā tika nodrošināta konfigurāciju pārvaldība, bet sestajā nodaļā ir aprakstīti projektā pielietotie kvalitātes nodrošināšanas mehānismi. Septītā nodaļa satur darbietilpības novērtējumu.

APZĪMĒJUMU SARAKSTS

Apzīmējums	Skaidrojums
TDD	Testu virzītā izstrāde (Test Driven Development)
TJ-PV	Tildes Jumis – personālvadība. SIA Tilde izstrādāts personālvadības un grāmatvedības mākoņrisinājums.
PA	Papildatvaļinājums
LESS	Programmēšanas valoda, kas tiek lietota, lai noteiktu tīmekļa lapas elementu stilu un attēlojumu. Atšķirībā no CSS nodrošina ērtāku nosacījumu veidošanu.
CSS	Programmēšanas valoda, kas tiek lietota, lai noteiktu tīmekļa lapas elementu stilu un attēlojumu.
Javascript	Augsta līmeņa, netipizēta, interpretējama programmēšanas valoda, ko darbina interneta pārlūkprogramma.
Angular	Google izstrādāts klienta puses tīmekļa lapu izstrādes ietvars.
.NET ietvars	Microsoft izstrādāts ietvars lietojumprogrammu izstrādei
XUnit	Ietvars vienībtestu rakstīšanai, izstrādājot programmatūru izmantojot .NET ietvaru.
Devexpress	Lietotāju saskarnes rīki, kas uzrakstīti Javascript valodā un ērti lietojami izstrādājot Angular lietotnes.
C#	Microsoft izstrādāta programmēšanas valoda
Entity Framework	Microsoft izstrādāts objektu-relāciju savietošanas ietvars.

1 Programmatūras prasību specifikācija

1.1 Ievads

1.1.1 Nolūks

Programmatūras prasību specifikācija (PPS) paredzēta, lai definētu atvaļinājumu uzskaites moduļa funkcionālās un nefunkcionālās prasības, kas ļautu veiksmīgi izstrādāt šo produkta sastāvdaļu. Dokuments paredzēts programmatūras izstrādātājam, turpmākajiem programmatūras uzturētājiem un programmatūras pasūtītājiem.

1.1.2 Darbības sfēra

Izstrādājamā programmatūra ir daļa no personāla pārvaldības sistēmas “Tildes Jumis - Personālvadība”. Atvaļinājumu uzskaites modulī jānodrošina iespēja izvēlēties, pēc kāda principa aprēķināt darbinieku ikgadējo atvaļinājumu un PA, kā arī jāspēj aprēķināt darbinieka uzkrāto ikgadējo atvaļinājumu un PA, ņemot vērā sistēmā ievadīto informāciju par attiecīgo darbinieku, tas ir, datumu, kad viņš sācis strādāt, kad bijis atvaļinājumā utt.

1.1.3 Saistība ar citiem dokumentiem

Šajā dokumentā apkopoto prasību projektējums ir aprakstīts programmatūras projektējuma aprakstā un šī projektējuma testēšanas plāns ir aprakstīts testēšanas dokumentācijā.

1.1.4 Pārskats

Šajā dokumentā iekļauts ievads, vispārējais apraksts, funkcionālo prasību apraksts lietotāju stātu formā, kā arī nefunkcionālo prasību apraksts.

1.2 Vispārējais apraksts

1.2.1 Produkta perspektīva

Atvaļinājumu uzskaites modulis ir sistēmas “Tildes Jumis - Personālvadība” (TJ-PV) sastāvdaļa, kas savukārt ir tīmekļa programma, ar kuras palīdzību tiek veikta

personāla un grāmatvedības pārvaldība. TJ-PV var izveidot uzņēmumu un pievienot tam darbiniekus, informāciju par tiem, pievienot uzņēmuma struktūrvienības, nodaļas, amatus un piesaistīt tos darbiniekam. Šajā sistēmā var uzskaitīt darbinieka nostrādāto laiku un aprēķināt darba samaksu. Ir iespējams arī reģistrēt dažādus notikumus, piemēram, kad darbinieks ir pieņemts darbā, kad viņš bijis atvaļinājumā, vai bijis slims. Šie reģistrētie notikumi tiek ņemti vērā aprēķinot darba samaksu un ir arī ļoti būtiski PPS aprakstītā projekta izstrādē, jo balstoties uz tiem, atvaļinājumu uzskaites modulis aprēķinās uzkrātos atvaļinājumus.

1.2.2 Produkta funkcijas

Atvaļinājumu uzskaites modulim jānodrošina šādu funkciju izpilde:

- 1) atvaļinājumu uzskaites iestatījumu automātiska izveidošana un iespēja tos rediģēt;
- 2) uzkrātā atvaļinājuma manuāla ievadīšana jeb korekcijas ievadīšana kā arī šī ievadītā uzkrājuma labošana un dzēšana;
- 3) uzkrāto atvaļinājumu un PA aprēķināšana lietotāja izvēlētam datumam;

1.2.3 Lietotāju raksturiezīmes

Lielākā daļa lietotāju ir grāmatveži, kuriem jau ir pieredze darbā ar grāmatvedības datorprogrammām, visbiežāk ar "Tildes Jumi". Šo sistēmu paredzēts lietot arī konkrētā uzņēmuma darbiniekiem, kuri varēs paši ievadīt datus, kas attiecas uz viņiem.

1.2.4 Vispārējie ierobežojumi

Sistēma ir jāizstrādā atbilstoši Latvijas Republikas likumdošanai, ņemot vērā darba likumu, kurā aprakstītas darbinieka tiesības uz atvaļinājumu. Modulim ir jābūt savietojamam ar TJ-PV sistēmu.

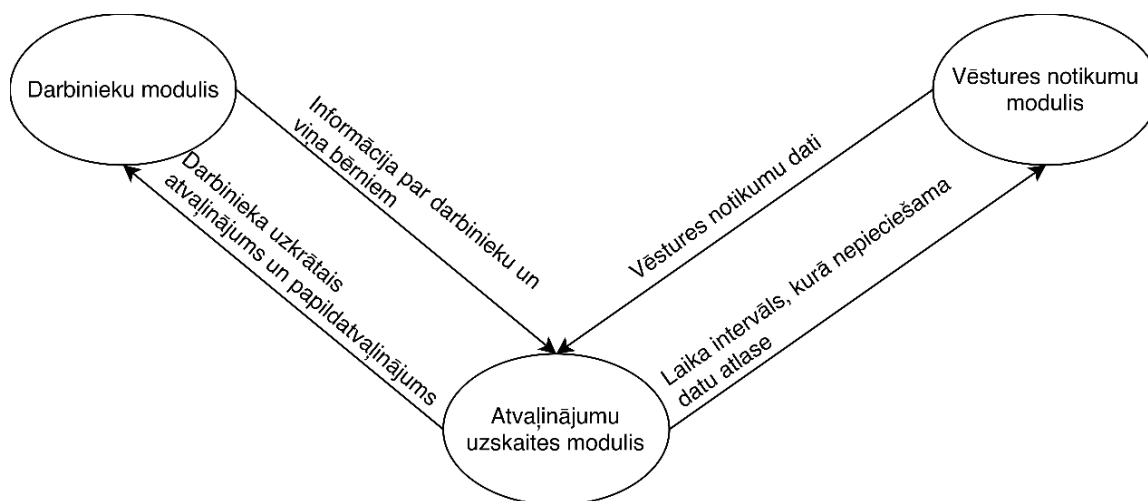
1.2.5 Pieņēmumi un atkarības

Programmatūras prasību specifikāciju ietekmē izmaiņas darba likumā.

1.3 Konkrētās prasības

1.3.1 Funkcionālās prasības

Šajā nodaļā aprakstītas atvaļinājumu uzskaites moduļa funkcijas lietotāju stāstu formā. Atvaļinājumu uzskaites modulis ir atkarīgs no vēstures notikumu moduļa. Vēstures notikumu modulī (1. att. **0. līmeņa Datu plūsmu diagramma**) tiek reģistrēti tādi darbinieku vēstures notikumi, kā “Pieņemts darbā”, kas apzīmē notikumu, kad darbinieks sācis darba attiecības ar uzņēmumu – no šī notikuma datuma darbiniekam sāk aprēķināt uzkrāto atvaļinājumu. Notikums “Atbrīvots no darba” apzīmē to, kad darbinieks pārtraucis darba attiecības ar uzņēmumu – tas ir notikums, ar kuru beidzas atvaļinājuma uzkrājuma aprēķināšana. Vēstures notikumu modulī tiek reģistrēts arī tas, kad darbinieks izmantojis atvaļinājumu. Darbiniekam ir iespējams izmantot dažāda veida atvaļinājumus, piemēram, parastu uzkrāto atvaļinājumu, PA, bērnu kopšanas atvaļinājumu, atvaļinājumu bez darba samaksas saglabāšanas. Šie visi notikumi tiek ņemti vērā aprēķinot uzkrātā atvaļinājuma un papildatvaļinājuma uzkrājumu.



1. att. **0. līmeņa Datu plūsmu diagramma**

1.3.1.1 Lietotāju stāsti

1.3.1.1.1 LS_1 Noklusēto iestatījumu izveidošana

Grāmatvedis un uzņēmuma vadītājs vēlas, lai katram darbiniekam tiktu automātiski rēķināts atvaļinājuma un papildatvaļinājuma uzkrājums, balstoties uz Latvijas Republikas likumiem. Pēc darba likuma noteikumiem, atvaļinājumu aprēķina

noklusētajos iestatījumos tiks ievadīti sekojoši dati: darbiniekam pienākas 28 kalendārās dienas garš apmaksāts atvaļinājums katru gadu. Ja darbinieks ir jaunāks par 18 gadiem, tad viņam pienākas 30 kalendārās dienas garš atvaļinājums. Laikā, par kuru pienākas atvaļinājums, neieskaita laiku, kad darbinieks pavadījis bērnu kopšanas atvaļinājumā. Šajā laikā ieskaita bezalgas atvaļinājumu līdz 4 nedēļām, bet neieskaita bezalgas atvaļinājumu virs 4 nedēļām gada laikā. Darbiniekam par 3 vai vairāk bērniem līdz 16 gadu vecumam pienākas 3 kalendāro dienu garš papildatvaļinājums (PA). 3 dienas pienākas arī par to, ka darbiniekam ir bērns ar invaliditāti līdz 18 gadu vecumam. Ja darbiniekam ir bērni līdz 14 gadu vecumam, viņam pienākas viena papildatvaļinājuma diena gadā. Ja darbinieka darbs saistīts ar īpašu risku, viņam pienākas 3 papildatvaļinājuma dienas gadā.

Akceptēšanas kritēriji – pievienotajiem darbiniekiem tiek pareizi aprēķināti uzkrātie atvaļinājumi un papildatvaļinājumi, neveicot nekādas izmaiņas sistēmas iestatījumos.

1.3.1.1.2 LS_2 Uzkrātā atvaļinājuma un papildatvaļinājuma manuāla ievadīšana konkrētā datumā

Ievadot darbinieku sistēmā, grāmatvedis vai uzņēmuma vadītājs vēlas norādīt darbinieka uzkrāto atvaļinājumu kalendārajās dienās, kā arī PA kalendārajās dienās norādot atsevišķu uzkrājumu par katru no iespējamajiem papildatvaļinājuma veidiem, kas aprakstīti darba likuma 151. pantā. Tas ir, par to, ka darbiniekam ir 3 vai vairāk bērnu vecumā līdz 16 gadiem vai bērns ar invaliditāti līdz 18 gadu vecumam, par to, ka darbiniekam ir bērni līdz 14 gadu vecumam, par to, ka darbinieka darbs saistīts ar īpašu risku, kā arī par citiem papildatvaļinājuma veidiem par ko vienojas darbinieks un darba devējs savā starpā. Tas nepieciešams, jo datu ievadītāji nevēlas reģistrēt sistēmā vēstures notikumus, kas saistīti ar atvaļinājuma aprēķinu sākot ar datumu, kad darbinieks sācis strādāt. Piemēram, ja darbinieks pieņemts darbā 1999. gadā, tad bez šī uzkrājuma ievadīšanas, pareizi aprēķināt uzkrāto atvaļinājumu varētu tikai tad, ja sistēmā ievadītu visus darbinieka vēstures notikumus, kas notikuši 18 gadu laikā. Lietotājam ir iespēja pievienot vairākus šādus ierakstus dažādos datumos. Turpmākos aprēķinus ietekmē tas uzkrājums, kas aktīvs uz vistālāko datumu.

Akceptēšanas kritēriji – lietotājs var veiksmīgi pievienot uzkrāto atvaļinājumu un PA kalendārajās dienās, kā arī norādīt, kurā dienā veikts šis aprēķins. Sākot ar ievadīto datu apstiprināšanas brīdi, sistēmas aprēķinātajam uzkrājumam jābūt pareizam, ja vien ir ievadīti pareizi dati.

1.3.1.1.3 LS_3 Noklusēto iestatījumu labošana

Grāmatvedis un uzņēmuma vadītājs vēlas, lai sistēmas iestatījumos varētu apskatīt un labot noklusētos iestatījumus, kas tiek ņemti vērā aprēķinot atvaļinājumu. Jābūt iespējai labot visus iestatījumus, kas aprakstīti lietotāja stāstā “LS_1”

Akceptēšanas kritēriji – grāmatvedis vai uzņēmuma vadītājs var sistēmas iestatījumos labot noklusētos atvaļinājumu aprēķināšanas iestatījumus, un, kad tie apstiprināti, tie tiek ņemti vērā veicot aprēķinu.

1.3.1.1.4 LS_4 Uzkrātā atvaļinājuma aprēķināšana darbiniekam

Darbinieks, grāmatvedis vai uzņēmuma vadītājs vēlas, lai varētu aprēķināt darbinieka uzkrāto atvaļinājumu norādot datumu, kuram jāveic aprēķins. Piemēram, var aprēķināt uzkrājumu šodienai vai aprēķināt to šī gada pirmajam janvārim. Aprēķinot uzkrāto atvaļinājumu kalendārajās dienās, jāņem vērā visi vēstures notikumi, kas reģistrēti sistēmā un ir saistīti ar atvaļinājumiem, tas ir, atvaļinājums jāsāk rēķināt no notikuma “Pieņemts darba”, jāpārēķina līdz notikumam “Atbrīvots no darba”, ja tāds notikums ir noticis pirms izvēlēta aprēķina datuma. Jāpārēķina visi izmantotie atvaļinājumi, kā arī jāpārēķina laiks, kas par kuru nepienākas atvaļinājums (bērnu kopšanas atvaļinājums, bezalgas atvaļinājums). Kad uzkrājums tiek aprēķināts, tas tiek saglabāts pie visiem konkrētā darbinieka uzkrājumiem. Vienā datumā nevar būt vairāki uzkrājumi.

Akceptēšanas kritēriji - Sistēma aprēķina uzkrāto atvaļinājumi atbilstoši konfigurācijai un ievadītajam uzkrājumam jeb labojumam. Piemēram, ja darbiniekam ir vēstures notikums “Pieņemts darbā” 13.04.2016 un viņš līdz 13.04.2017 nav izmantojis nevienu atvaļinājuma dienu, kā arī ievadītais uzkrājums uz 13.04.2016 ir 0 dienas, tad kā aprēķina datumu norādot 13.04.2017., darbiniekam vajadzētu parādīt 28 uzkrātās atvaļinājuma dienas.

1.3.1.1.5 LS_5 Uzkrātā papildatvaļinājuma aprēķināšana

Darbinieks, grāmatvedis vai uzņēmuma vadītājs vēlas, lai sistēma aprēķinātu darbinieka uzkrāto PA datumā, kuru tas norādījis. PA jāaprēķina par katru darba likumā noteikto kritēriju atsevišķi, ja sistēmas iestatījumos atzīmēts, ka šis kritērijs tiek izmantots. To, cik dienas par katru kritēriju gadā pienākas darbiniekam, nosaka sistēmas iestatījumos norādītās vērtības.

Akceptēšanas kritēriji – Norādot aprēķina datumu, lietotājs saņem atskati par uzkrāto PA kalendārajās dienās par katru kritēriju, kas sistēmas iestatījumos atzīmēts, kā izmantots.

1.3.1.1.6 LS_6 Ievadītā atvaļinājuma un papildatvaļinājuma uzkrājuma labošana.

Grāmatvedis vai uzņēmuma vadītājs vēlas, lai varētu labot LS_1 aprakstīto ievadīto uzkrājumu, ja veicot LS_1 aprakstīto darbību lietotājs ir kļūdījies. Sistēmā jābūt iespējai nospiegt pogu “Labot ievadīto uzkrājumu”, tādējādi parādot formu, kurā ir iepriekšējā ievadīšanas reizē ievadītie dati, ar iespēju tos labot un apstiprināt labojumu.

Akceptēšanas kritēriji – Nospiežot pogu “Labot ievadīto uzkrājumu”, lietotājam tiek parādīta forma, kurā var ievadīt visus LS_1 aprakstītos datus. Šī forma ir aizpildīta ar tiem datiem, kuri tika ievadīti pirmajā reizē. Nospiežot pogu “Saglabāt”, sistēma saglabā ievadītos datus un turpmāk aprēķina uzkrāto atvaļinājumu un PA ņemot vērā šos datus.

1.3.1.1.7 LS_7 Saglabāto uzkrājumu attēlošana

Lietotājs vēlas, lai varētu pārskatāmi apskatīt visus ievadītos un aprēķinātos uzkrātos atvaļinājumus. Ir jāizveido tabula, kurā ir attēloti visi šie ievadītie un aprēķinātie uzkrājumi, kā arī iespēja tos dzēst un, ja uzkrājums ievadīts manuāli, rediģēt, bet, ja aprēķināts automātiski – pārrēķināt. Tabulā vienā rindā jāattēlo datums, kurā veikts aprēķins, kā arī visas iespējamās uzkrājuma sadaļas, kas atzīmētas, kā izmantotas uzkrātā atvaļinājuma aprēķināšanas iestatījumos. Ievadītie ieraksti jāsakārto pēc datuma augošā secībā.

Akceptēšanas kritēriji – lietotājs var atvērt tabulu, kurā tiek attēloti visi iepriekš ievadītie un aprēķinātie uzkrājumi. Ja uzkrājums ir ievadīts manuāli, tad šo ierakstu var rediģēt, bet, ja uzkrājums ir automātiski aprēķināts, tad šo ierakstu var pārrēķināt.

1.3.1.1.8 LS_8 Manuāli ievadīto uzkrājumu dzēšana

Lietotājam jābūt iespējai dzēst manuāli ievadītos uzkrājumus. Ja šis uzkrājums ir visvēlāk aprēķinātais un tas tiek ņemts vērā, kā korekcija, tad, dzēšot šo uzkrājumu, ir jāatrod nākamais vēlākais uzkrājums, ja tāds eksistē, un šis turpmāk jāņem vērā kā korekcija.

Akceptēšanas kritēriji – dzēšot ievadītu uzkrājumu, tiek atrasts tuvākais iepriekš ievadītais uzkrājums, ja tāds eksistē, un turpmāk tiek ņemts vērā, ka korekcija.

1.3.2 Nefunkcionālās prasības

1.3.2.1 Ārējās saskarnes prasības

1.3.2.1.1 NP_1 Atbilstība TJ-PV

Atvaļinājumu uzskaites moduļa ārējā saskarne ir jāizstrādā tādā pašā stilā, kā izstrādāta TJ-PV sistēma un tās apakš moduļi. Visām tabulām, virsrakstiem, ievades laukiem un paziņojumiem jābūt tādā pašā stilā, kā pārējos TJ-PV sistēmas moduļos. Modulis jāizstrādā tā, lai lietotājs varētu ērti izmantot visus sistēmas moduļus, ievērojot vienādus lietošanas paradumus.

2 Programmatūras projektējuma apraksts

2.1 Ievads

2.1.1 Nolūks

Šajā dokumentā aprakstīta programmatūras prasību specifikācijā aprakstītā programmatūras produkta funkcionālo un nefunkcionālo prasību realizācija.

2.1.2 Darbības sfēra

Izstrādājamā programmatūra ir daļa no personāla pārvaldības sistēmas “Tildes Jumis - Personālvadība”. Atvaļinājumu uzskaites modulī jānodrošina iespēja izvēlēties, pēc kāda principa aprēķināt darbinieku ikgadējo atvaļinājumu un PA, kā arī jāspēj aprēķināt darbinieka uzkrāto ikgadējo atvaļinājumu un PA, ņemot vērā sistēmā ievadīto informāciju par katru darbinieku, tas ir, datumu, kad viņš sācis strādāt, kad bijis atvaļinājumā utt.

2.1.3 Saistība ar citiem dokumentiem

Šajā dokumentā tiek projektētas prasības, kas ir aprakstītas programmatūras prasības specifikācijā. Šī projektējuma realizācijas testēšanas plāns ir aprakstīts testēšanas dokumentācijā.

2.1.4 Dokumenta pārskats

Dokuments satur ievadu, dekompozīcijas aprakstu, datubāzes projektējumu un nefunkcionālo prasību realizācijas aprakstu.

2.2 Dekompozīcijas apraksts

Uzkrāto atvaļinājumu aprēķina modulis satur šādas entītijas, entītiju atribūtus un metodes (2. att. **Metožu attiecību datu plūsmu diagramma**):

- Uzkrātā ikgadējā atvaļinājuma aprēķina rezultāts (VacationResult)
Šī entītija netiek glabāta datubāzē, tā paredzēta, lai apstrādātu datus, kas saņemti pēc datubāzes pieprasījuma izpildes.

Atribūti
Darbinieka ID (EmployeeId) – vesels skaitlis
Atvaļinājuma dienu skaits (VacationDays) – decimālskaitlis
Metodes
Aprēķināt uzkrāto atvaļinājumu (CalculateVacation) – atgriež iepriekš aprakstīto entītiju, parametros saņem darbinieka ID, aprēķina sākuma un beigu datumu.
Aprēķināt dienu skaitu, par kuru nepienākas atvaļinājums (GetExcludedDays) – parametros saņem darbinieka ID, atgriež dienu skaitu, par kurām nepienākas atvaļinājums.
Atrod datumu (GetVacationBrakepoints), kas ietekmē darbinieka uzkrātā atvaļinājuma aprēķinu, tas ir, datumu, kad darbiniekam paliek 18 gadi, ja šis datums ir intervālā starp datumu, kad darbinieks sācis strādāt un datumu, kad veikts aprēķins.

- Uzkrātā papildatvaļinājuma aprēķina rezultāts (ExtraVacationResult)

Šī entītija netiek glabāta datubāzē, tā paredzēta, lai apstrādātu datus, kas saņemti pēc datubāzes pieprasījuma izpildes.

Atribūti
Darbinieka ID (EmployeeId) – vesels skaitlis
Papildatvaļinājuma dienu skaits par 3 un vairāk bērniem līdz 16 gadu vecumam (ForChildrenUnder16) – decimālskaitlis
Papildatvaļinājuma dienu skaits, kas pienākas par mazāk nekā 3 bērniem līdz 14 gadu vecumam (ForChildrenUnder14) - decimālskaitlis
Papildatvaļinājuma dienu skaits, kas pienākas par bērnu invalīdu līdz 18 gadu vecumam (ForDisabledUnder18) - decimālskaitlis
Papildatvaļinājuma dienu skaits, kas pienākas par to, ka darbinieks nodarbināts amatā, kas saistīts ar īpašu risku (ForHighRisk) - decimālskaitlis
Papildatvaļinājuma dienu skaits, pēc koplīguma starp darbinieku un darba devēju (ForOther) - decimālskaitlis
Metodes
Aprēķināt uzkrāto PA (CalculateExtraVacation) – atgriež iepriekš aprakstīto entītiju, parametros saņem darbinieka ID, aprēķina sākuma un beigu datumu.
GetExcludedDays – skatīt VacationResult metodes
Noteikt, vai darbinieks atbilst kritērijiem, par kuriem pienākas PA (GetExtraVacationData) – parametros saņem darbinieka ID numuru un aprēķina sākuma un beigu datumu, atgriež kritēriju sarakstu ar pazīmi atbilst vai neatbilsti pie katra.
Atrod zīmīgus datumus (CalculateExtraVacationBrakepoints), kas ietekmē darbinieka PA aprēķināšanu, piemēram, kad piedzimuši bērni, kad tie sasnieguši 17 gadu vecumu utt. atbilstoši aprēķina iestatījumiem.

- Uzkrātā atvaļinājuma aprēķina iestatījumi

Atribūti
Uzņēmuma ID (CompanyId) – vesels skaitlis
Atvaļinājumu dienu skaits darbiniekiem, kas vecāki par 18 gadiem (VacationDays) – vesels skaitlis
Atvaļinājumu dienu skaits darbiniekiem, kas jaunāki par 18 gadiem (VacationDaysUnder18) – vesels skaitlis
Laikā, par kuru pienākas atvaļinājums, neieskaitīt bērna kopšanas atvaļinājumu (ExcludeParentalLeave) – jā/nē
Laikā, par kuru pienākas atvaļinājums, neieskaitīt bezalgas atvaļinājumu līdz 4 nedēļām (ExcludeUnpaidUnder4) – jā/nē
Laikā, par kuru pienākas atvaļinājums, neieskaitīt bezalgas atvaļinājumu virs 4 nedēļām (ExcludeUnpaidOver4) – jā/nē
Piešķirt PA par 3 un vairāk bērniem līdz 16 gadu vecumam (UseChildrenUnder16) – jā/nē
Piešķiramo papildatvaļinājuma dienu skaits gadā, ja darbiniekam ir 3 vai vairāk bērnu līdz 16 gadu vecumam (ForChildrenUnder16) – vesels skaitlis
Piešķirt PA par 2 un mazāk bērniem līdz 14 gadu vecumam (UseChildrenUnder14) – jā/nē
Piešķiramo papildatvaļinājuma dienu skaits gadā, ja darbiniekam ir 2 vai mazāk bērnu līdz 14 gadu vecumam (ForChildrenUnder14) – vesels skaitlis
Piešķirt PA par bērniem invalīdiem līdz 18 gadu vecumam (UseDisabledUnder18) – jā/nē
Piešķiramo papildatvaļinājuma dienu skaits gadā, ja darbiniekam ir bērns ar invaliditāti līdz 18 gadu vecumam (ForDisabledUnder18) – vesels skaitlis
Piešķirt PA par to, ka darbinieks nodarbināts amatā ar īpašu risku (UseHighRisk) – jā/nē
Piešķiramo papildatvaļinājuma dienu skaits gadā, ja darbinieks ir nodarbināts amatā ar īpašu risku (ForHighRisk) – vesels skaitlis
Piešķirt PA par noteikumiem, kas atrūnāti koplīgumā (UseOther) – jā/nē

Piešķiramo papildatvaļinājuma dienu skaits gadā pēc koplīguma (ForOther) – vesels skaitlis
Metodes
Izveidot noklusētos atvaļinājuma aprēķina iestatījumus (CreateDefaultVacationPolicy) – parametros saņem uzņēmuma ID.
Labot noklusētos atvaļinājuma aprēķina iestatījumus (EditVacationPolicy) – parametros saņem iepriekš aprakstīto entīti ar veiktajiem labojumiem.

- Darbinieka uzkrātā atvaļinājuma dati (EmployeeVacation)

Atribūti
Darbinieka ID (EmployeeId) – vesels skaitlis
Datums, kurā veikts aprēķins (CalculationDate) – datums
Uzkrātais atvaļinājums (VacationDays) – decimālskaitlis
Uzkrātais PA par bērniem līdz 14 gadu vecumam (ForChildrenUnder14) – decimālskaitlis
Uzkrātais PA par bērniem līdz 16 gadu vecumam (ForChildrenUnder16) – decimālskaitlis
Uzkrātais PA par bērniem ar invaliditāti līdz 18 gadu vecumam (ForDisabledUnder18) – decimālskaitlis
Uzkrātais PA par darbu amatā ar īpašu risku (ForHighRisk) – decimālskaitlis
Uzkrātais PA par citiem nosacījumiem (ForOther) – decimālskaitlis
Metodes
Aprēķināt uzkrāto atvaļinājumu un PA (GetEmployeeVacation) – parametros saņem darbinieka ID un datumu, kurā jāaprēķina uzkrājums. Tiek izsauktas metodes CalculateVacation un CalculateExtraVacation, kuru rezultāti tiek apkopoti šajā entītijā un atgriezti sistēmai. Ja datubāzē ir saglabāta uzkrājuma korekcija konkrētam darbiniekam, tad tā jāatņem no iegūtā rezultāta.
Manuāli ievadīta uzkrātā atvaļinājuma saglabāšana (AddManualVacation). Ja datu bāzē neeksistē uzkrājuma aprēķina korekcija ar jaunāku datumu, tad šis ievadītais uzkrājums jāpadod korekcijas metodei (AddCorrection), lai aprēķinātu jaunu korekciju.
Sameklēt datumu, kad darbinieks sācis strādāt un beidzis strādāt (GetEmployeeWorkingDates) – saņem parametros darbinieka ID, atgriež datumu, kad darbinieks sācis strādāt, un datumu, kad beidzis strādāt, ja tāds eksistē.

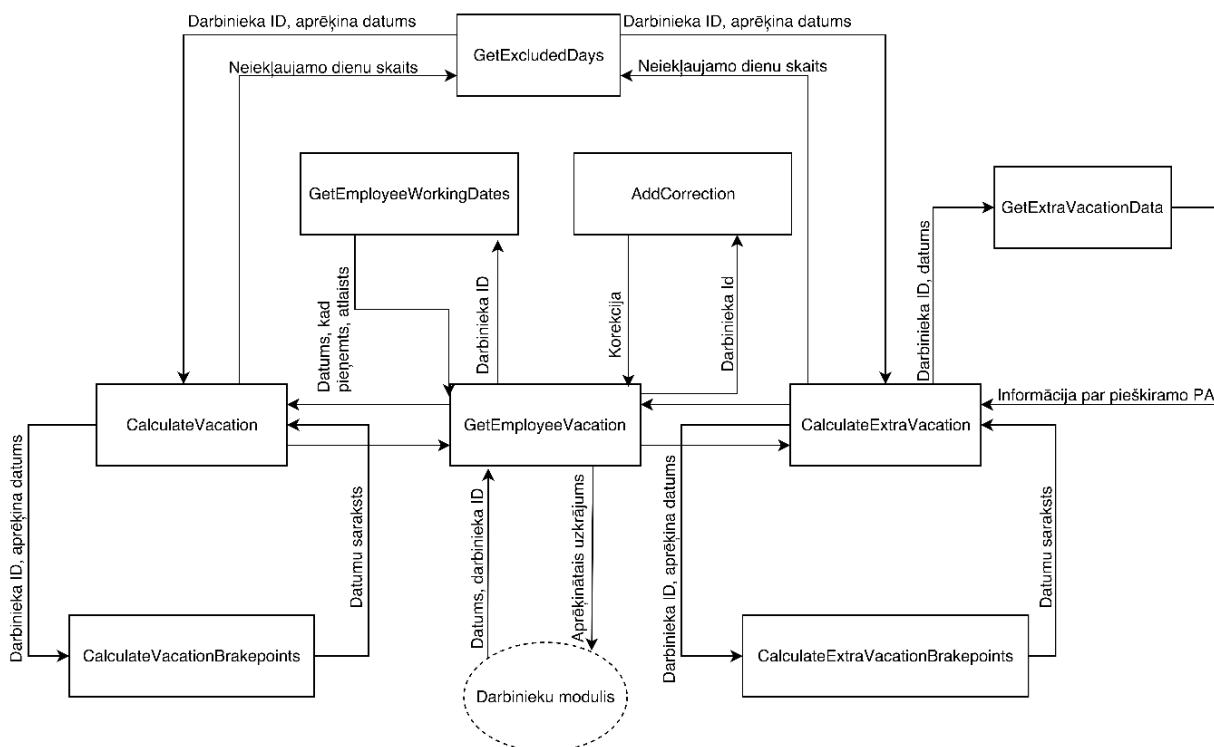
- Darbinieka uzkrātā atvaļinājuma un PA korekcija (EmployeeVacationCorrection)

Atribūti
Darbinieka ID (EmployeeId) – vesels skaitlis
Datums, kurā ievadīta korekcija (CorrectionAdded) - datums
Kalendārās atvaļinājuma dienas datumā, no kura rēķināta korekcija (VacationDaysOnDate) - decimālskaitlis
Kalendāro atvaļinājuma dienu korekcija, kas tiek rēķināta no potenciāla aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (Correction) - decimālskaitlis
Uzkrātā papildatvaļinājuma dienu skaits par bērniem līdz 14 gadu vecumam norādītajā datumā (ForChildrenUnder14OnDate) – decimālskaitlis
Kalendāro papildatvaļinājuma dienu korekcija par bērniem līdz 14 gadu vecumam, kas tiek rēķināta no potenciāla aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (ForChildrenUnder14Correction) - decimālskaitlis
Uzkrātā papildatvaļinājuma dienu skaits par bērniem līdz 16 gadu vecumam norādītajā datumā (ForChildrenUnder16OnDate) – decimālskaitlis
Kalendāro papildatvaļinājuma dienu korekcija par bērniem līdz 16 gadu vecumam, kas tiek rēķināta no potenciāla aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (ForChildrenUnder16Correction) - decimālskaitlis
Uzkrātā papildatvaļinājuma dienu skaits par bērniem ar invaliditāti līdz 18 gadu vecumam norādītajā datumā (ForDisabledUnder18OnDate) – decimālskaitlis
Kalendāro papildatvaļinājuma dienu korekcija par bērniem ar invaliditāti līdz 18 gadu vecumam, kas tiek rēķināta no potenciāla aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (ForDisabledUnder18Correction) - decimālskaitlis
Uzkrātā papildatvaļinājuma dienu skaits par darbu amatā ar īpašu risku norādītajā datumā (ForHighRiskOnDate) – decimālskaitlis
Kalendāro papildatvaļinājuma dienu korekcija par darbu amatā ar īpašu risku, kas tiek rēķināta no potenciāli aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (ForDisabledUnder18Correction) - decimālskaitlis
Uzkrātā papildatvaļinājuma dienu skaits par citiem nosacījumiem norādītajā datumā (ForOtherOnDate) – decimālskaitlis

Kalendāro papildatvaļinājuma dienu korekcija par citiem nosacījumiem, kas tiek rēķināta no potenciāli aprēķinātā atvaļinājuma dienu skaita atņemot faktisko uzkrājumu konkrētā datumā (ForOtherCorrection) - decimālskaitlis

Metodes

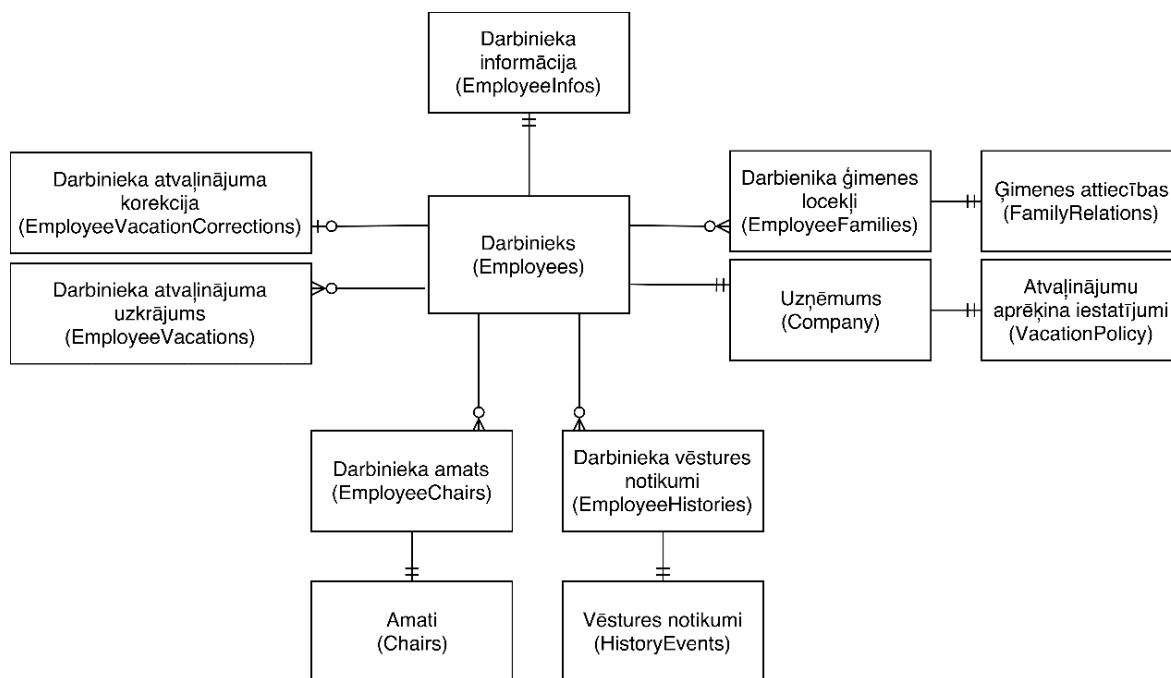
Korekcijas aprēķināšana un pievienošana vai labošana (AddCorrection) – parametros saņem darbinieka atvaļinājuma entītiju (EmployeeVacation) ar ievadītu uzkrājumu un datumu, kurā šis uzkrājums bijis spēkā. Tālāk tiek aprēķināts atvaļinājuma potenciālais uzkrājums šajā datumā katrā atvaļinājuma un papildatvaļinājuma pozīcijā. Šīs vērtības tiek atņemtas no parametros saņemtajām faktiskajām vērtībām un tā tiek iegūta uzkrājuma korekcija.



2. att. Metožu attiecību datu plūsmu diagramma

2.3 Datubāzes projektējums

Šajā nodaļā aprakstīts izstrādājamā moduļa entītiju attēlojums datubāzē. Datubāzes diagrammās (3. att. **Konceptuālais datubāzes ER modelis**) tiks iekļautas arī citas entītijas, kas nepieciešamas, lai aprēķinātu uzkrāto atvaļinājumu un PA darbiniekam. Šīs entītijas jau ir izveidotas citos TJ-PV moduļos un šeit netiks aprakstītas.



3. att. Konceptuālais datubāzes ER modelis

2.3.1 Datubāzes tabulas

Datubāze tika realizēta izmantojot *Microsoft Entity Framework Core* pēc *Code-first* principa, tas ir, vispirms tika izveidotas modeļu klases valodā *C#* un tad ar *Entity Framework* tās tikai pārvērstas par MS-SQL servera datubāzes tabulām (7. att.).

2.3.1.1 Tabula “Atvaļinājuma aprēķina iestatījumi (VacationPolicy)”

Kolonna	Datu tips	Cita informācija
EmployeeId	integer	primārā atslēga, automātiska numerācija
CompanyId	integer	ārējā atslēga no tabulas Companies (šajā dokumentā nav aprakstīta)
VacationDays	decimal	
VacationDaysUnder18	decimal	
ExcludeParentalLeave	boolean	
ExcludeUnpaidUnder4	boolean	
ExcludeUnpaidOver4	boolean	

UseChildrenUnder16	boolean	
ForChildrenUnder16	decimal	
UseChildrenUnder14	boolean	
ForChildrenUnder14	decimal	
UseDisabledChildrenUnder18	boolean	
ForDisabledUnder18	decimal	
UseHighRisk	boolean	
ForHighRisk	decimal	
UseOther	boolean	
ForOther	decimal	

2.3.1.2 Tabula “Darbinieka atvaļinājuma korekcija (EmployeeVacationCorrection)”

Kolonna	Datu tips	Cita informācija
Id	integer	primārā atslēga, automātiska numerācija
EmployeeId	integer	ārējā atslēga no tabulas “Employees”
CompanyId	integer	ārējā atslēga no tabulas “Companies”
CorrectionAdded	date	
Correction	decimal	
VacationDaysOnDate	decimal	
ForChildrenUnder16OnDate	decimal	
ForChildrenUnder16Correction	decimal	
ForChildrenUnder14OnDate	decimal	
ForChildrenUnder14Correction	decimal	
ForDisabledUnder18OnDate	decimal	
ForDisabledUnder18Correction	decimal	

ForHighRiskOnDate	decimal	
ForOtherCorrection	decimal	
ForOtherOnDate	decimal	

2.3.1.3 Tabula “Darbinieka atvaļinājuma uzkrājums (EmployeeVacation)”

Kolonna	Datu tips	Cita informācija
Id	integer	primārā atslēga, unikāls identifikators, automātiska numerācija
EmployeeId	integer	ārējā atslēga no tabulas “Employees”
CompanyId	integer	ārējā atslēga no tabulas “Companies”
CalculationDate	date	
CalculatedOn	date	
VacationDays	decimal	
ForChildrenUnder16	decimal	
ForChildrenUnder14	decimal	
ForDisabledUnder18	decimal	
ForOther	decimal	
ForHighRisk	decimal	

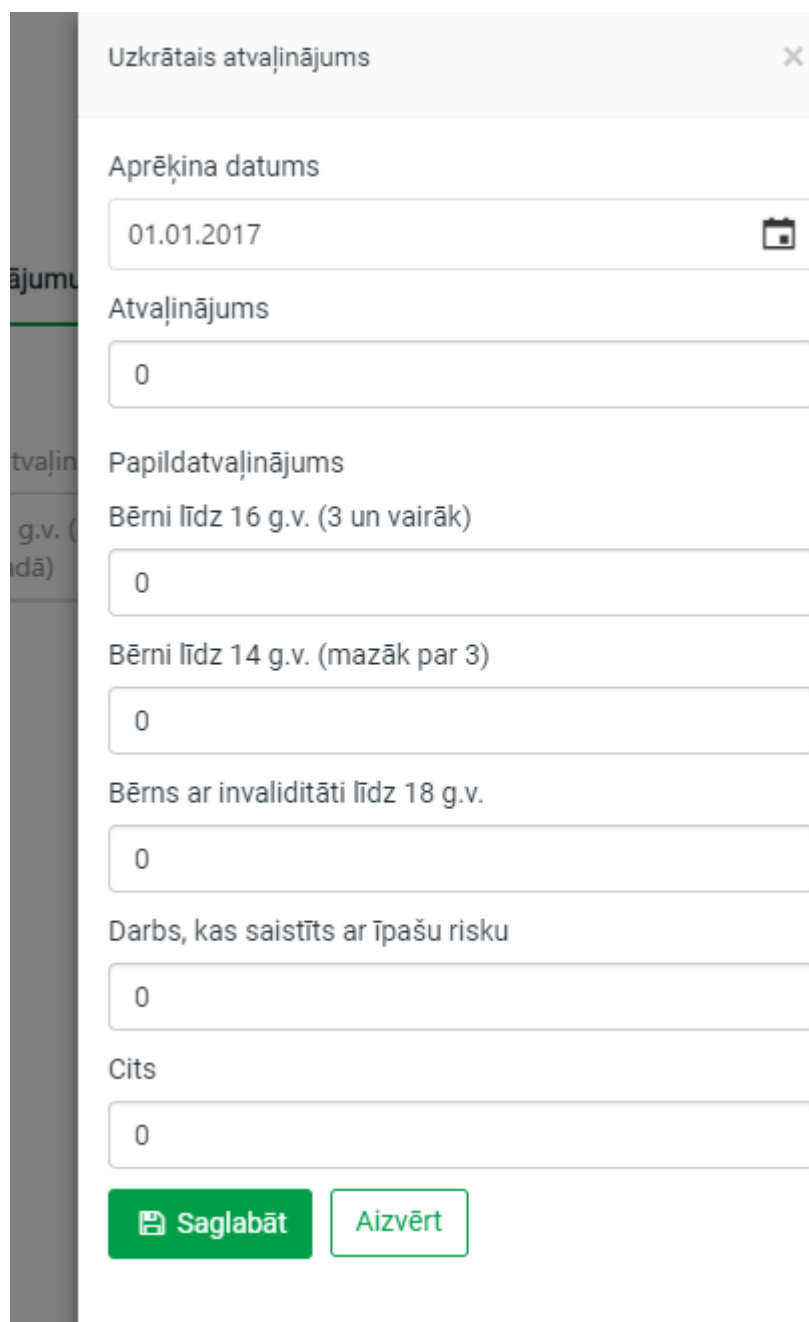
2.4 Ekrānformu projektējums

Atvaļinājumu uzkrājumu saraksts tiek attēlots tabulas veidā (4.att) ar iespējām pievienot uzkrājumu, nospiežot pogu “Pievienot uzkrājumu”, un meklēt tabulā, ja tajā

Uzkrājums uz ↑	Uzkrātais atvaļinājums (28 dienas gadā)	Uzkrātais papildatvaļinājums			Dzēst	Pārrēķināt/ Labot
		Bērni līdz 16 g.v. (3 dienas gadā)	Bērni līdz 14 g.v. (1 dienas gadā)	Īpašs risks (3 dienas gadā)		
01.01.2017	0	0	0	0		
25.05.2017	11.05	0	0	0		

4. att. Uzkrāto atvaļinājumu attēlojums

attēloti daudz ierakstu. Pēc “Pievienot uzkrājumu” pogas nospiešanas, tiek piedāvāta izvēle – “Aprēķināt” un “Ievadīt uzkrājumu”.



Uzkrātais atvaļinājums

Aprēķina datums

01.01.2017

Atvaļinājums

0

Papildatvaļinājums

Bērni līdz 16 g.v. (3 un vairāk)

0

Bērni līdz 14 g.v. (mazāk par 3)

0

Bērns ar invaliditāti līdz 18 g.v.

0

Darbs, kas saistīts ar īpašu risku

0

Cits

0

Saglabāt Aizvērt

5. att. Uzkrātā atvaļinājuma manuālas ievadišanas forma

Nospiežot pogu “Ievadīt uzkrājumu”, tiek izsaukta manuāla uzkrājuma pievienošanas funkcija. Manuāli ievadīt uzkrāto atvaļinājumu var vienkāršā formā (5.att.), kas tiek attēlota modālā logā. Ievadītos datus var apstiprināt ar pogu “Saglabāt”, bet aizvērt formu, ja lietotājs nevēlas ievadīt datus, var ar pogu “Aizvērt”.

Uzkrātā atvaļinājuma aprēķina iestatījumu forma ir ievietota izvēršamās cilnēs, kuras, nepieciešamības gadījumā var aizvērt. Šīs ir akordeona tipa cilnes, kas ļauj tās ērti izvietot vienā logā. (6.att.), kur atsevišķi atdalīti atvaļinājuma un papildatvaļinājuma aprēķina iestatījumi. Papildatvaļinājumu sadaļā – ja kāds no aprēķina kritērijiem sistēmā tiek atzīmēts, kā neizmantots, tad tam attiecīgais dienu skaita ievades lauks tiek atspējots, bet iepriekš ievadītā vērtība ir jāsaglabā.

▲ Atvaļinājuma uzkrāšana

<p>Kalendāro atvaļinājuma dienu skaits gadā</p> <p>Darbiniekiem, kuri vecāki par 18 gadiem <input style="width: 60px;" type="text" value="28"/></p> <p>Darbiniekiem, kuri jaunāki par 18 gadiem <input style="width: 60px;" type="text" value="30"/></p>	<p>Laiks, par kuru nepienākas ikgadējais atvaļinājums</p> <p><input checked="" type="checkbox"/> Bērna kopšanas atvaļinājums</p> <p><input type="checkbox"/> Bezalgas atvaļinājums līdz 4 nedēļām</p> <p><input checked="" type="checkbox"/> Bezalgas atvaļinājums virs 4 nedēļām</p>
--	---

▲ Papildatvaļinājuma uzkrāšana

Nosacījumi, par kuriem tiek piešķirts papildatvaļinājums un piešķiramo dienu skaits

<input checked="" type="checkbox"/> Bērni līdz 16 g.v. (3 un vairāk)	<input style="width: 60px;" type="text" value="3"/>	<input checked="" type="checkbox"/> Darbs saistīts ar īpašu risku	<input style="width: 60px;" type="text" value="3"/>
<input checked="" type="checkbox"/> Bērni līdz 14 g.v. (līdz 3)	<input style="width: 60px;" type="text" value="1"/>	<input type="checkbox"/> Pēc kopīguma	<input style="width: 60px;" type="text" value="0"/>
<input type="checkbox"/> Bērni ar invaliditāti līdz 18 g.v.	<input style="width: 60px;" type="text" value="3"/>		

6. att. Uzkrātā atvaļinājuma aprēķina iestatījumu forma

2.5 Nefunkcionālo prasību realizācija

Lai panāktu to, ka moduļa ārējā saskarne tiek izstrādāta vienā stilā ar TJ-PV sistēmu, tajā jālieto globālie *LESS* faili, kā arī *CSS* un *JavaScript* bibliotēka *Bootstrap*, kas nodrošina saskarnes ērtu pielāgošanu dažādām ierīcēm ar dažādiem ekrānu izmēriem. Tabula datu attēlošanai tiek realizēta, izmantojot *Devexpress* lietotāju saskarnes rīkus.

3 Testēšanas dokumentācija

3.1 Ievads

3.1.1 Nolūks

Testēšanas dokumentācija izveidota, lai aprakstītu to, kā tika testēta programmatūras prasību specifikācijā aprakstītā un programmatūras projektējuma aprakstā realizētā programmatūra.

3.1.2 Saistība ar citiem dokumentiem

Šajā nodaļā tiek testēta programmatūra, kuras prasības aprakstītas programmatūras prasību specifikācijā un projektējuma apraksts aprakstīts programmatūras projektējuma aprakstā.

3.2 Testēšanas apraksts

Tā kā projekts tika izstrādāts pēc Testu virzītās izstrādes principa (Test driven development - TDD), galvenās funkcionalitātes vienībtesti tika izveidoti vēl pirms funkcionalitātes programmēšanas. Vienībtesti tika izstrādāti, izmantojot atvērtā pirmkoda “XUnit” vienībtestēšanas rīku, kas paredzēts.NET ietvaram. Pirms vienībtestu rakstīšanas tika izdomāti testēšanas scenāriji, kas atbilst reāliem dzīves piemēriem. Lietotāju saskarnes funkcionalitāte tika testēta manuāli.

Testēšanas žurnālā apkopoti šo testu apraksti un izpildes rezultāti. Ja rezultāta kolonnā ir '+', tests izpildīts veiksmīgi – ja '-', tad tests nav bijis veiksmīgs. Tabulā ir apkopoti katra testa pēdējie darbības rezultāti.

3.3 Testēšanas žurnāls

3.3.1 Automātiskie vienībtesti

Testa nosaukums	Skaidrojums	Iznākums
Testing_Employee_Vacation_Calculation_If_Gets_18_While_Working	Pārbauda, vai brīdī, kad darbiniekam paliek 18 gadu, tiek samazināts	+

	atvaļinājuma dienu skaits gadā.	
Testing_Vacation_Calculation_If_Working_One_Year	Pārbauda, vai tiek pareizi aprēķināts uzkrājums, ja darbinieks strādā pilnu gadu.	+
Testing_Correction_Calculation_Method	Pārbauda korekcijas aprēķināšanas metodi	+
Testing_Extra_Vacation_Calculation_For_High_Risk	Pārbauda, vai tiek piešķirtas papildatvaļinājuma dienas, ja darbinieks nodarbināts amatā, kas saistīts ar īpašu risku.	+
Testing_Extra_Vacation_Calculation_For_Children_Under_16	Pārbauda, vai darbiniekam pareizi piešķir papildatvaļinājuma dienas, ja tam ir 3 vai vairāk bērnu līdz 16 gadu vecumam.	+
Testing_Extra_Vacation_Calculation_For_Children_Under_14	Pārbauda, vai darbiniekam pareizi piešķir papildatvaļinājuma dienas, ja tam ir līdz 2 bērniem, kuri nav sasnieguši 14 gadu vecumu.	+
Testing_Extra_Vacation_Calculation_For_Disabled_Child	Pārbauda, vai darbiniekam pareizi tiek piešķirtas papildatvaļinājuma dienas par to, kad viņam ir bērns invalīds līdz 18 gadu vecumam.	+
Testing_Breakpoint_Finding	Pārbauda, vai sistēma pareizi atrod zīmīgus datumus, kuros mainās uzkrātā atvaļinājuma aprēķins.	+

Testing_Hiring_Firing_Date_Getter	Pārbauda, vai sistēma pareizi atrod datumus, kad darbinieks pieņemts darbā un kad no darba atbrīvots.	+
-----------------------------------	---	---

3.3.2 Manuālie testi

Testa nosaukums	Skaidrojums	Iznākums
Testing_same_date_saving	Uzkrājuma manuāla ievadīšana datumā, kurā jau ir ievadīts vai aprēķināts uzkrājums. Pozitīvs iznākums, ja nevar pievienot.	+
Testing_show_only_used_columns	Attēlojot saglabātos uzkrājumus tabulā, jāredz tikai tās kolonnas, kas iestatījumos atzīmētas, kā izmantotas.	+
Testing_order_by_date	Datiem atskaites tabulāt jāsakārtojas pēc datuma.	+
Testing_catch_invalid_date	Aprēķinot uzkrājumu jāievada korekts datums. Ja tas nav izdarīts, jāizvada kļūdu paziņojums un jāpārtrauc darbība.	+
Testing_calculate_or_edit	Saglabāto uzkrājumu tabulā jābūt iespējai labot ierakstu, ja tas ievadīts manuāli un pārrēķināt to, ja aprēķināts automātiski.	+

4 Projekta organizācija

Projekts tika organizēts pēc testu virzītās izstrādes metodoloģijas (Test driven development - TDD), kur pēc prasību savākšanas tika izstrādāti galvenās funkcionalitātes vienību testi, kas tika izmantoti, lai pakāpeniski attīstītu testējamās vienības. Šāds izstrādes princips nodrošina to, ka programmatūrai uzreiz ir gatavi vienību testi, kas pēc katrām izmaiņām programmas kodā, ļauj nekavējoties pārbaudīt, vai kods aizvien precīzi izpilda komandas un iegūst nepieciešamo rezultātu. Šis princips ļauj arī labāk uzturēt kodu pēc tam, kad to pārņems cits programmētājs.

TDD katras jaunas funkcionalitātes ieviešana sākas ar testu rakstīšanu. Šim testam jāpārbauda jaunās funkcionalitātes darbība. Tā kā tests tiek uzrakstīts pirms programmatūras, tas sākumā vienmēr ir neveiksmīgs. Šis noteikums liek programmētājam sākt izpētīt prasības vēl pirms koda rakstīšanas. Kad jaunie testi ir uzrakstīti un pārbaudīti, ka tie ir neveiksmīgi, programmētājs var sākt attiecīgās funkcionalitātes rakstīšanu un attiecīgi to testēt ar izveidotajiem testiem, kamēr tie ir veiksmīgi. Kad tiek sasniegts tas, ka testi izpildās veiksmīgi, programmētājs var sākt uzlabot un pārveidot kodu, līdz tas kļūst *ideāls*.

5 Konfigurācijas pārvaldība

Konfigurācijas pārvaldība, izstrādājot attiecīgo programmatūru, tika nodrošināta ar *Microsoft Team Foundation Server* (TFS). Visas programmatūras kodā veiktās izmaiņas tika augšupielādētas serverī, kas aprīkots ar TFS. Augšupielādētās izmaiņas tika komentētas, lai nepieciešamības gadījumā varētu ātrāk atrast nepieciešamo izmaiņu kopu un salīdzināt attiecīgās kopas programmatūras kodu ar pašreizējo kodu. TFS tika izmantots arī darbu plānošanā, veidojot uzdevumus un atzīmējot tos, kā izdarītus.

6 Kvalitātes nodrošināšana

Lai nodrošinātu kvalitatīvu projekta izstrādi, tika izstrādāta programmatūras prasību specifikācija un programmatūras projektējuma apraksts, ievērojot valsts noteiktos standartus. Izstrādājot programmatūru, tika ievērots labais programmēšanas stils, tas ir, saturīgi nosaukti programmu faili, klašu un funkciju nosaukumi, kā arī mainīgie. Programmas pirmkods tika komentēts, lai to vēlāk varētu vieglāk saprast.

Izstrādei tika izmantota *Visual Studio 2017 Enterprise* izstrādes vide. Servera puses programmatūra tika veidota izmantojot *.NET Core* tīmekļa lapu ietvaru kopā ar *Entity Framework Core*, bet klienta puses daļa tika veidota izmantojot izstādes brīdī jaunāko pieejamo *Angular* ietvaru. Lai nodrošinātu ātrāku datu apstrādi un uzkrājuma aprēķinu, dažas metodes tikai īstenotas kā SQL funkcijas. Programmatūra tika izstrādāta pēc TDD principa, kas ievērojami samazina kļūdu iespējamību un būtiski atvieglo programmatūras turpmāku uzturēšanu.

7 Darbietilpības novērtēšana

Lai novērtētu to, cik darbietilpīgs ir šis projekts un vai tā apjoms atbilst kvalifikācijas darba prasībām, tika veikts darbietilpības novērtējums pēc procesu novērtēšanas metodes. Tika saplānoti darāmie darbi un novērtēti trīs veidos – optimistiski, reālistiski, pesimistiski. Attiecīgie darbi tika vērtēti cilvēkstundās, ņemot vērā darba autora pieredzi un iemaņas programmēšanā. Katra darba novērtējuma rezultāts tika iegūts pēc formulas (*optimistiskais* + 4 * *reālistiskais* + *pesimistiskais*) / 6.

Darbs	Novērtējums (stundas)			Rezultāts (stundas)
	Optimistiskais	Reālistiskais	Pesimistiskais	
Prasību apkopošana	4	8	12	8
Likumu izpēte	2	4	8	4
Dokumentācijas rakstīšana	90	130	160	128
Vienībtestu izstrāde	70	80	90	80
Aprēķina iestatījumu funkcionalitātes izstrāde	16	32	64	35
Datubāzes projektējuma izstrāde un realizācija	8	16	20	13
Galvenās funkcionalitātes izstrāde	80	120	200	127
Lietotāju saskarnes izstrāde	60	80	100	80
Lietotāju saskarnes testēšana	3	6	9	6
			Kopā	481 stunda

Novērtējot visus darāmos darbus, tika aprēķināts, ka tie aizņems 481 stundu. Tā kā vienā personmēnesī ir 160 stundas, tikai iegūti 3 personmēneši, kas arī atbilst kvalifikācijas darba nepieciešamajam apjomam.

8 Secinājumi

Kvalifikācijas darba rezultātā ir izstrādāts funkcionējošs TJ-PV moduļa programmatūras kods, kas tiek reāli izmantots TJ-PV sistēmā gan uzkrātā atvaļinājuma aprēķināšanai, gan šīs sistēmas citos moduļos, piemēram, aprēķinot algas, programmatūras prasību specifikācija, programmatūras projektējuma apraksts un testēšanas dokumentācija. Izstrādātā moduļa funkcionalitāte atbilst izvirzītajām prasībām.

Projekta izstrāde, ieskaitot prasību savākšanu, dokumentācijas izstrādi, programmēšanu un testēšanu, aizņēma trīs personmēnešus.

Realizējot projektu, tika secināts, ka universitātē iegūtās zināšanas ir tikai pats pamats, lai varētu apgūt padziļinātākas prasmes programmēšanā, kuras reāli ir noderīgas darba tirgū. Izstrādājot programmatūru, autors apguva daudz jaunu tīmekļa programmēšanas tehnoloģiju, kā arī pamatīgi nostiprināja universitātē iegūtās zināšanas. Šī projekta izstrāde bija pirmā darba autora pieredze salīdzinoši lielā un ar funkcionalitāti bagātā projektā, kas pamatīgi atšķīrās no iepriekš veiktajiem projektiem.

Turpmāk izstrādāto moduli varētu attīstīt ar papildu funkcionalitāti, kas ļautu katram uzņēmumam vēl vairāk pielāgot atvaļinājumu piešķiršanas mehānismus.

9 Izmantotā literatūra

1. Beck, K. Test-Driven Development by Example (2003)
2. Angular dokumentācija, <https://angular.io/docs/ts/latest/>
3. XUnit dokumentācija, <https://xunit.github.io/#documentation>
4. LR Darba likums, <https://likumi.lv/doc.php?id=26019>
5. Entity Framework dokumentācija, [https://msdn.microsoft.com/en-us/library/ee712907\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/ee712907(v=vs.113).aspx)

10 Pielikumi

10.1 Koda fragmenti

```
[Fact]
public void Testing_Extra_Vacation_Calculation_For_Children_Under_16()
{
    decimal result = 0;
    decimal expected = 10.70M;
    int empId = db.Employees
        .First(x => x.CompanyId==context.SelectedCompany).Id;
    var from = vacationRepo.GetEmployeeWorkingDates(empId).Result[0];
    var forWaiting = familyRepo.GetFamilyRelations().Result;
    //Lai sagaida datu ievietošanu
    int familyRelationId =
        db.FamilyRelations
            .First(x => x.IsChild == true &&
                x.Company.Id == context.SelectedCompany).Id;
    var company = db.Companies.First(x => x.Id == context.SelectedCompany);

    //Izveido darbiniekam 3 bērņus
    EmployeeFamily child1 = new EmployeeFamily();
    child1.EmployeeId = empId;
    child1.Name = child1.Surname = "test";
    child1.Birthday = new DateTime(2002, 9, 9);
    child1.FamilyRelationId = familyRelationId;
    child1.Company = company;
    db.EmployeeFamilies.Add(child1);

    EmployeeFamily child2 = new EmployeeFamily();
    child2.EmployeeId = empId;
    child2.Name = child2.Surname = "test";
    child2.Birthday = new DateTime(2004, 6, 10);
    child2.FamilyRelationId = familyRelationId;
    child2.Company = company;
    db.EmployeeFamilies.Add(child2);

    EmployeeFamily child3 = new EmployeeFamily();
    child3.EmployeeId = empId;
    child3.Name = child3.Surname = "test";
    child3.Birthday = new DateTime(2009, 8, 10);
    child3.FamilyRelationId = familyRelationId;
    child3.Company = company;
    db.EmployeeFamilies.Add(child3);
    db.SaveChanges();

    result += vacationRepo
        .CalculateExtraVacation(empId, from,
            new DateTime(2025, 9, 9)).Result.ForChildrenUnder16;

    Assert.Equal(expected, result);
}
```

```

/// <summary>
/// Aprēķina darbinieka uzkrāto atvaļinājumu padotajā laika periodā.
/// </summary>
/// <param name="EmployeeId"></param>
/// <param name="DateFrom"></param>
/// <param name="DateTo"></param>
/// <returns>
/// Uzkrāto atvaļinājumu darbiniekam
/// </returns>
public async Task<VacationResults>
CalculateVacation(int EmployeeId, DateTime DateFrom, DateTime DateTo)
{
    DateTime vacationBrakePoint =
    await GetVacationCalculationBrakepoints(EmployeeId, DateFrom, DateTo);
    List<DateTime> dateList = new List<DateTime>();
    dateList.Add(DateFrom);
    dateList.Add(DateTo);
    if (vacationBrakePoint.Year >= 19)
        dateList.Add(vacationBrakePoint);
    dateList.Sort();

    VacationResults result = new VacationResults();
    result.EmployeeId = EmployeeId;
    result.CalculatedVacation = 0;

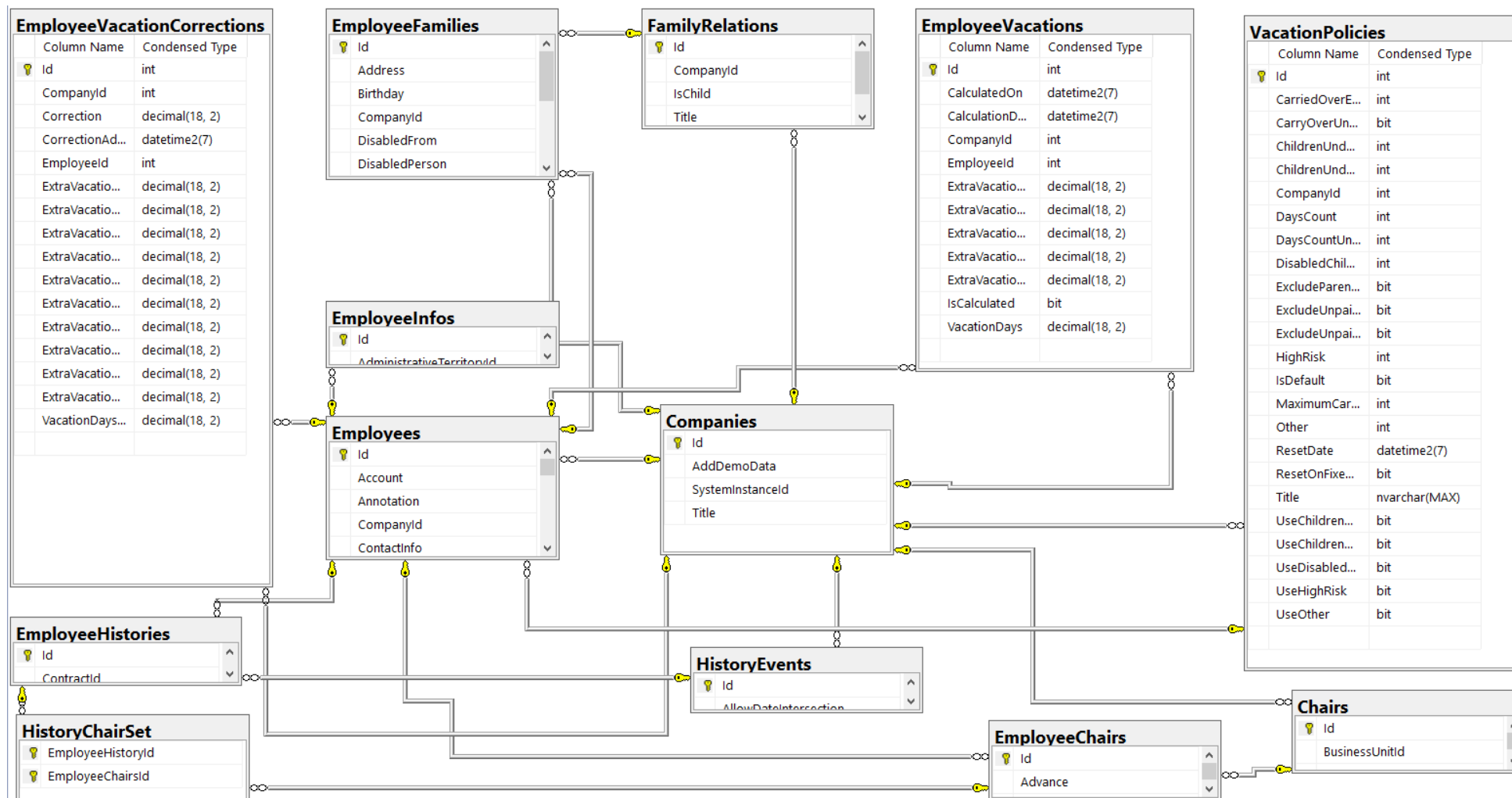
    //Atvaļinājuma aprēķins starp zīmīgajiem datumiem, kas maina aprēķina rez.
    for (int count = 0; count < dateList.Count - 1; count++)
    {
        var from = dateList[count];
        var to = dateList[count + 1];
        string DateFromStr = $"{from.Date.ToString("yyyy-MM-dd")}";
        string DateToStr = $"{to.Date.ToString("yyyy-MM-dd")}";
        var sql = String.Format(SqlStringForVacation, DateFromStr, DateToStr,
            $"={EmployeeId}", AuthorizedUser.SelectedCompany);

        var data = await customADO.GetVacationData(sql);
        result.CalculatedVacation += data[0].CalculatedVacation;
    }
    result.CalculatedVacation = Math.Round(result.CalculatedVacation, 2);

    return result;
}

```

10.2 Datubāzes realizācijas ER modelis



7. att. Datubāzes realizācijas ER modelis

Kvalifikācijas darbs „Prombūtnes uzskaites moduļa izstrāde personālvadības sistēmā”
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: Rihards Mozaļevskis _____ .05.2017.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs M. Dat. Matīss Rikters _____ .05.2017.

Recenzents: M. Dat. Inguna Pede

Darbs iesniegts 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): _____