

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Lietišķo datorsistēmu institūts
Lietišķo datorzinātņu katedra

Sergejs TERENCEVS
2. kursa RDGDB1 grupas students
(stud. apl. Nr. 061RDB140)

BIBLIOTĒKAS INFORMĀCIJAS SISTĒMAS „ALISE” PORTĀLA IZSTRĀDE

Ražošanas prakses pārskats

Students: _____
(paraksts, pārskata nodošanas datums)

Prakses vadītājs: Māris Krivtežs, SIA „Tieto Latvia” vecākais programmētājs, M.sc.ing.

(paraksts, datums)

RTU prakses koordinators: Pāvels Rusakovs, asoc. prof., Dr.sc.ing. _____
(paraksts, pārskata pārbaudes datums)

Prakses pārskats aizstāvēts 2012. gada ”____” jūnijā

Rīga – 2012

Anotācija

Prakses ietvaros tika izstrādāta informācijas sistēma, realizējot projekta ietvaros nostādītos uzdevumus. Uzdevumos ietilpa „BIS ALISE” informatīva klienta portāla izstrāde, veicot sistēmas arhitektūras projektēšanu, vadoties pēc programmatūras projektēšanas paņēmieniem un šabloniem, lietojuma drošības un aizsardzības pasākumu realizēšana, paredzot aizsargāto klientu sadaļu, adaptīvas sistēmas izstrādāšanas principu apgūšana, nodrošinot iespēju lietotājiem uzturēt sistēmu, kā arī veikt sistēmas ieviešanas un uzturēšanas darbus, apgūt uzņēmuma organizāciju, darbības virzienus un struktūras.

Izstrādāta sistēma ir uzlikta produkcijas vidē un pilda uzdevuma ietvaros noteiktas funkcijas. Sistēmas izstrāde ir veikta .NET vidē, pielietojot MS SQL Server datubāzi. Sistēmas izstrādē ir izmantotas ASP.NET MVC, ASP.NET WebForms, N2 CMS, Dapper tehnoloģijas. Sistēmas izstrāde ir veikta C# valodā.

Darbs satur 28 lappuses, 27 attēlus un 3 pielikumus. Literatūras saraksts satur 11 informācijas avotus.

SATURS

IEVADS	5
1. UZŅĒMUMA RAKSTUROJUMS	6
1.1. Uzņēmuma vēsture.....	6
1.1.1. „Tieto” vēsture.....	6
1.1.2. „Tieto Latvia” vēsture	7
1.2. Organizācijas struktūra un darbības virzieni	8
1.3. Darba pienākumi un atbildība	10
2. IZSTRĀDĀTAS SISTĒMAS RAKSTUROJUMS.....	12
2.1. Risināmie uzdevumi un sistēmas apraksts	12
2.2. Sistēmas arhitektūras projektēšana.....	13
2.3. Uzdevumu risināšana	17
2.3.1. Lietojuma struktūras izstrāde.....	17
2.3.2. Sistēmas domēna un datu pieejas līmeņa izstrāde	21
2.3.3. Aizsargātas klientu sadaļas izstrāde	23
2.3.4. Administrācijas moduļa izstrāde	27
2.4. Sistēmas ieviešana un uzturēšana.....	28
SECINĀJUMI	31
LITERATŪRA	32
PIELIKUMI.....	33
LIETOTI SAĪSINĀJUMI.....	35

IEVADS

Studijas programmā noteikta ražošanas prakse ir īstenota uzņēmumā „Tieto Latvia”. Prakses programmas īstenošana ilga 20 nedēļas, nostrādājot 800 darba stundas. Prakses pienākumu pildīšana uzņēmumā ir uzsākta līdz ar 2012. gada 30. janvārī un tika veikta līdz 2012. gada 15. jūnijam, ieskaitot.

Ražošanas prakses mērķis bija nostiprināt studenta studiju procesā iegūtas teorētiskais zināšanas, praktiskas prasmes un iemaņas programmēšanas inženierim atbilstošo uzdevumu risināšanā. Mērķa sasniegšanai studentam bija jāiepazīstas ar programmatūras sistēmu izstrādes un uzturēšanas metodēm, programmatūras izstrādes vidēm, rīkiem un tehnoloģijām, uzņēmuma lietotām operētājsistēmām, datubāzu un datortīklu tehnoloģijām, uzņēmuma darba organizāciju, darba plānošanu, vadīšanu un pārvaldību. Uzdevuma saraksts atrodas pārskata 2. lappusē.

Prakses pārskatā pirmā daļā ir sniegts ieskats uzņēmuma vēsturē, ir apskatītas uzņēmuma struktūrvienības un darbības virzieni, kā arī aprakstīti praktikanta darba pienākumi un atbildība. Pārskatā otrā daļā ir sniegts izstrādātas sistēmas raksturojums, ir aprakstīts uzdevumu risināšanas process, ar to saistītie pētījumi, izmantotas tehnoloģijas un iegūtie rezultāti. Atsauksme par praksi atrodas 1. pielikumā. Lietotu saīsinājumu saraksts atrodas 2. pielikumā.

1. UZŅĒMUMA RAKSTUROJUMS

SIA „Tieto Latvia” ir IT pakalpojumu uzņēmums Latvijā, kas piedāvā informācijas tehnoloģiju risinājumus dažādās tautsaimniecības un biznesa jomās. SIA „Tieto Latvia” ir „Tieto” filiāle Latvijā, kas līdz ar vairākām Ziemeļeiropas, Āzijas un ASV valstīm veido vienoto „Tieto” organizāciju.

1.1. Uzņēmuma vēsture

SIA „Tieto Latvia” uzņēmuma vēsture ir saistīta ar „Tieto Corporation” organizācijas attīstību. Šajā sadaļā tiek apskatītas gan „Tieto Latvia”, gan „Tieto” uzņēmumu vēstures.

1.1.1. „Tieto” vēsture

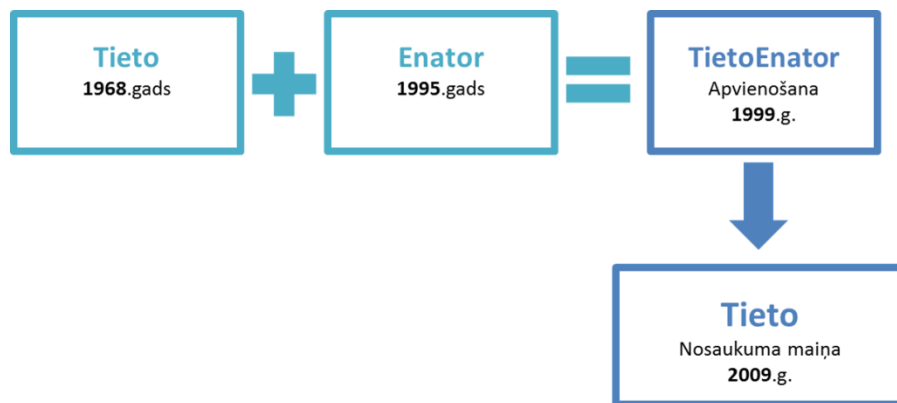
Pastāvoša „Tieto” uzņēmuma pirmsākumi tiek meklēti 1968.gadā, kurā Espo, Somijā, tika nodibināta organizācija „Teitotehdas Oy” [1]. Pirmajos pastāvēšanas gados uzņēmums uzturēja un attīstīja IT sistēmas. Turpmākajā desmitgadē uzņēmuma klientu bāze paplašinājās un papildus lieldatoru apkalpošanai un programmatūras izstrādei uzņēmums pievērsās arī personālo datoru sistēmām un programmatūrai.

Deviņdesmitajos gados uzņēmums piedzīvoja strauju izaugsmi, iegādājoties citus uzņēmumus, kā arī izmantojot apvienošanas un stratēģiskās partnerības ar citiem nozares uzņēmumiem. 1996. gadā uzņēmums nostiprināja savu pozīciju telekomunikāciju segmentā, iegādājoties uzņēmumu „Avancer” [1].

Šajos gados rādās arī uzņēmums „Enator”. Uzņēmums tika dibināts 1995. gadā, kad Zviedrijas korporācija „Celsius AB” apvienoja savas IT nodaļas [2]. Turpmākos gados „Enator” atdalījās no „Celsius AB” organizācijas un nostiprināja savu ietekmi valsts sektorā, pārņemot vairākus uzņēmumus.

1999. gadā notika „Tieto” un „Enator” uzņēmumu apvienošanas. Apvienošanas rezultātā rādās jauns uzņēmums „TietoEnator”, kas pēdējo desmit gadu laikā, IT industrijai globalizējoties, ir nostiprinājis savas pozīcijas starptautiskajā tirgū.

2009. gadā uzņēmums pieņēma jaunu nosaukumu „Tieto”. Uzņēmuma attīstības vēsture ir attēlota 1.1. attēlā.



1.1. att. „Tieto” attīstības vēsture [3]

1.1.2. „Tieto Latvia” vēsture

Uzņēmuma „Tieto Latvia” pirmsākumi tiek meklēti 1989. gadā Latvijas Universitātes ietvaros, kad atsevišķa darbinieku grupa izstrādāja sistēmu „COMPAS” Latvijas Universitātes bibliotēkas bibliotekāro procesu automatizācijai [4]. Pēc trim gadiem tika izstrādāta pirmā bibliotēku informācijas sistēmas „BIS ALISE” versija.

1994. gadā rādās uzņēmums „IT ALISE”, kas tika izveidots darbiniekiem nodaloties no Latvijas Universitātes Bibliotēku Automatizācijas nodaļas un kļūdāmi par patstāvīgo struktūrvienību.

Turpmākos gados uzņēmums izstrādāja un attīstīja jaunākas „BIS ALISE” sistēmas versijas. Tā 1995. gadā tika izstrādāta sistēmas versija, kas darbojas FoxPro un DOS vidē, bet 2000. gadā tika paredzēta sistēmas versija Windows videi [4]. No 2005. gadā uzņēmums izstrādā sistēmas Web versijas. Uz doto brīdī „BIS ALISE” ietver sevī divas sistēmas:

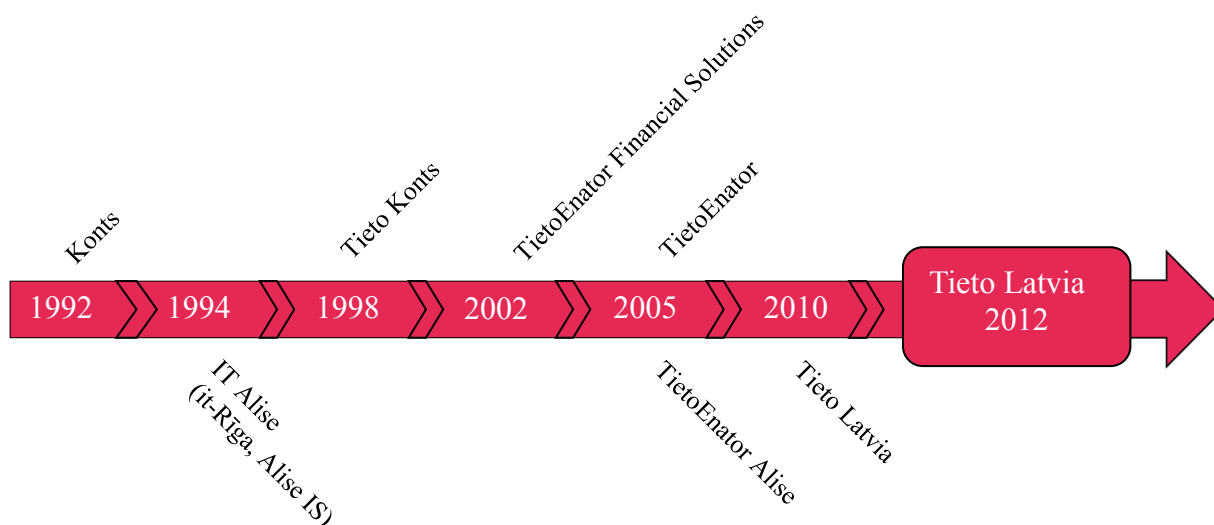
- „ALISE” – iekšējo datu vadības sistēma, piemēram, izņemto grāmatu reģistrēšanai un soda sankciju noteikšanai.
- „WebPAC” – ārēja sistēma, kas kalpo par katras atsevišķas bibliotēkas mājas lapu, piedāvājot iespēju aplūkot un pasūtīt noteikto literatūras avotu, uzzināt vairāk bibliotēku u.t.t.

2002. gadā rādās uzņēmums „Softshark”, kas turpmāk tika pārdēvēts par „TietoEnator Financial Solutions” [5]. Uzņēmums tika dibināts kā „TietoEnator” filiāle Baltijas valstīs un piedāvāja dažādus risinājumus banku apkalpošanas un finanšu jomā. Šādi risinājumi bija saistīti ar maksājuma karšu datu apstrādes sistēmas izstrādi un ieviešanu, kā arī dažādu IT

pakalpojumu nodrošināšanu bankām. Dažus gadus vēlāk uzņēmums pārņēma nosaukumu „TietoEnator”.

2005. gadā „IT ALISE” pievienojas klāt „TietoEnator” korporācijai un mainīja nosaukumu uz „TietoEnator Alise”. 2010. gadā notika abu „TietoEnator” pārstāvniecības uzņēmumu saliedēšana, kuras rezultātā, apvienojoties Latvijas uzņēmumiem „TietoEnator” un „TietoEnator Alise”, rādās jauna kompānija „Tieto Latvia”.

Uzņēmuma attīstības vēsture ir attēlota 1.2. attēlā.



1.2. att. „Tieto Latvia” attīstības vēsture

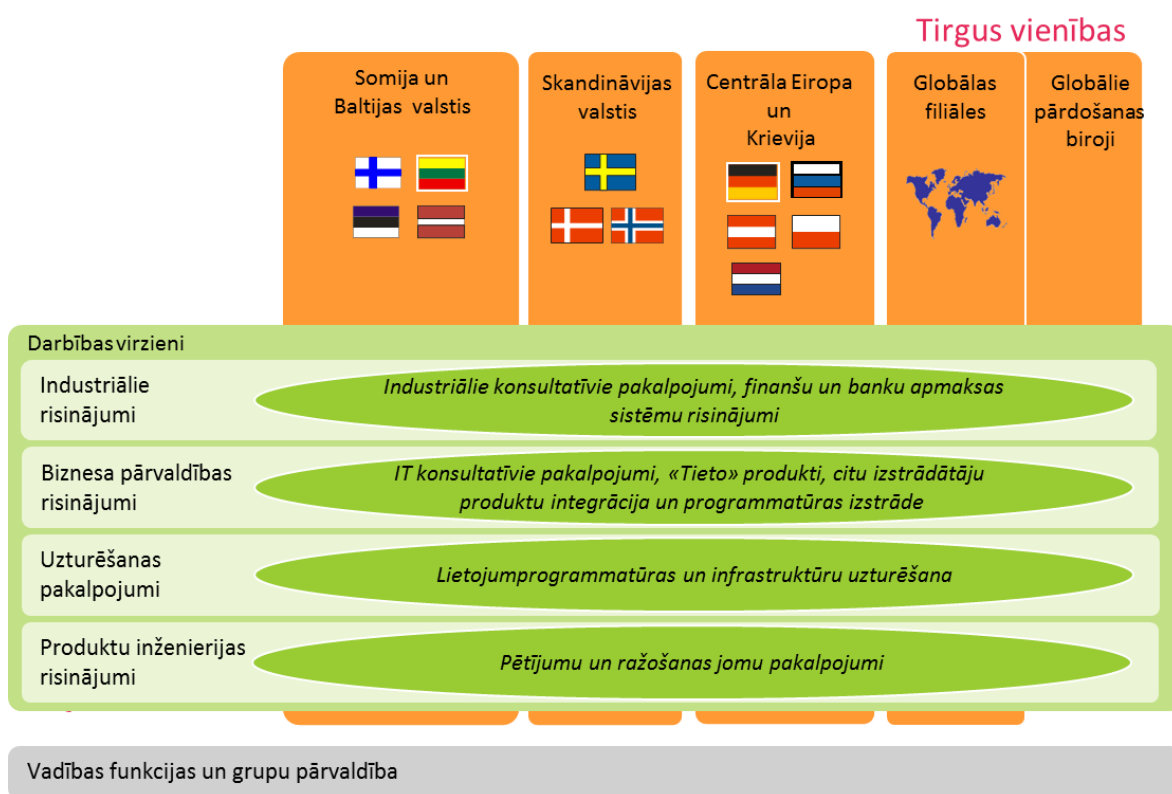
1.2. Organizācijas struktūra un darbības virzieni

„Tieto” galvenie darbības virzieni ir [7]:

- Industriālie risinājumi – ietver klientu biznesa procesu automatizācijas risinājumus, kuri ir specifiski īpašo nozaru vajadzībām. Šajā kategorijā ietilpst karšu personalizācijas risinājumi, maksāšanas pakalpojumu un finanšu sistēmu izstrāde, klientu konsultēšana, kā arī risinājumu integrācijas darbs.
- Biznesa pārvaldības risinājumi – kategorijā tiek ietverti vispārīgie risinājumi, kas saistās ar lietojumprogrammatūras izstrādi, produktu ieviešanu un integrāciju. Šis virziens paredz arī ERP un CRM sistēmu integrācijas risinājumu izstrādi.
- Uzturēšanas pakalpojumi – paredz izstrādāto un ieviesto sistēmu uzturēšanas darbus, kas saistīti ar defektu novēršanu, sistēmu uzlabošanu un attīstību.

- Produktu inženierijas risinājumi – kategorijā ietilpts projektēšanas, izstrādes un uzturēšanas risinājumi klientu lietotiem programmatūras produktiem. Šādi risinājumi tiek piemēroti mobilajām iekārtām, telekomunikāciju tīkliem, automobiļu ražošanas un pētniecības nozaru jomām.
- Vadības funkciju un grupu pārvaldības pakalpojumi – nodrošina finanšu pakalpojumus „Tieto” korporācijas uzņēmumiem.

„Tieto” korporācijas galvenie darbības virzieni un to tirgus sadalījuma segments ir attēlots 1.3. attēlā.



1.3. att. „Tieto” darbības modelis [6]

Attēlā ir redzams, ka „Tieto” tiek izdalīti četri galvenie tirgus vienību segmenti. Baltijas valstis un Somija ir viena no primāram IT pakalpojumu un risinājumu nodrošināšanas vadlīnijām. Dotās valstīs tiek izstrādāti risinājumi valsts sektora, mežsaimniecības, finanšu pakalpojumu, mazumtirdzniecības u.c. jomās. Skandināvijas „Tieto” vienība pozicionējas uz Zviedrijas, Norvēģijas un Dānijas valstu tirgu. Krievijas, Centrālās un Austrumu Eiropas tirgus vienība ir trešā primāra organizācijas vadlīnija, kas darbojas visos uzņēmuma darbības jomās. Globālā „Tieto” vienībā ietilpst Indijas, Ķīnas, ASV, Malaizijas, Filipīnas, Kanādas

u.c. valstis, kuru tendence ir globāla tirgus pozīciju nostiprināšana. Indijas un Āzijas valstis galvenokārt pozicionējas uz produktu inženierijas un uzturēšanas pakalpojumu darbības virzienu vadlīnijām.

Balstoties uz korporācijas darbības virzieniem, uzņēmumā „Tieto Latvia” tiek izdalītas sekojošas struktūrvienības [6]:

- „Financial Services Latvia” – iekļauj produktu un programmatūras izstrādes, piegādes, klientu atbalsta, kvalitātes vadības un pārdošanas nodaļas. Dotai struktūrvienībai ir raksturīgs industriālo risinājumu darbības virziens.
- „ERP, CRM and Integration Services” – ietver Microsoft ERP, ārvalstu un Baltijas EBS, pārskatu un pašapkalpošanās risinājumu, kā arī vietējo un „izaicinošo” projektu nodaļas. Dotas struktūrvienības darbības virziens ir biznesa procesu pārvaldība.
- „Global Shared Service Centre” – struktūrvienība satur vairākas nodaļas, kuras nodarbojas ar uzņēmuma finanšu lietām, analītiku un atskaitēm. Tai ir raksturīgs vadības funkciju un pārvaldības darbības virziens.
- „Managed Services Latvia” – pilda darbstaciju uzturēšanu, lietotāju tiesību administrēšanu, datortehnikas sagādi un remontu. Dota struktūrvienībā ietilpst administratoru grupa. Struktūrvienībai ir raksturīgs uzturēšanas pakalpojumu darbības virziens.

Uzņēmums satur arī mārketinga, personālas daļas un iestādes vadības nodaļas. Šīs nodaļas ir atbildīgas par biroja administratīvo, komandējumu un personāla administratīvo jautājumu risināšanu, kā arī atbild par pasākumu rīkošanu, sponsorēšanas aktivitātēm, uzņēmuma tēlu un suvenīru sagādi.

1.3. Darba pienākumi un atbildība

Pārskata autors uzņēmumā uzsāka strādāt pirms prakses īstenošanas, piesakoties programmētāja vakancei. Uzsākot darbu jaunajā darba vietā tiku iepazīstināts ar uzņēmuma darbības modeli, vēsturi, struktūrvienībām un bieži lietojamām iekšējam sistēmām padarīta darba uzskatei, tehnisko resursu iepirkuma pieprasījumu reģistrēšanai, sava darbinieka profila informācijas uzturēšanai, servisa atbalsta pieteikumu reģistrēšanai u.t.t. Darbojoties programmatūras izstrādātāja amatā, pārskata autora primārā kompetence ietilpst

programmatūras risinājumu izstrāde un esošo sistēmu uzturēšana, kā arī uzņēmuma uzstādīto normatīvu ievērošana, t.i., padarīta darba uzskaites aizpildīšana, profila atjaunošana u.t.t.

Pirmajās darba dienās praktikants tika piesaistīts klāt „IT ALISE” darbinieku grupa un tika pakļauts noteikta projekta vadītājam. Darbinieku grupa nodarbojas ar „BIS ALISE” produktu uzturēšanu un jaunāko versiju attīstību. Prakses gaitā esmu izstrādājis „BIS ALISE” sistēmas klientu informatīvo portālu, kurš tika ieskaitīts kā mans prakses darbs.

Bez esoša projekta esmu piesaistīts arī citiem projektiem. Šādi projekti ir:

- „Valvecert” – somu projekts, kas ir paredzēts „Metso” organizācijas svārstu un virzuļu rūpniecības struktūrvienībai.
- „LATLIT” – projekts, kas tiek lietots vides aizsardzības un reģionālās attīstības ministrijas ietvaros, lai piešķirtu Eiropas finansējumu Latvijā vai Lietuvā topošiem projektiem.

Darbs pie šiem projektiem paredz sistēmu uzturēšanu, defektu novēršanu un izmaiņu ieviešanu.

Darbojoties projektos, praktikants bieži ir spiests sazināties ar klientiem vai nu ar projektu vadītāju, lai apspriestu risināmo uzdevumu stāvokli un darbus, kurus būtu jāveic. Sistēmu defekti un nepieciešamie uzlabojumi tiek pieteikti projektu vadības sistēmā „JIRA”, nosakot tiem atbilstošo prioritāti.

Strādājot „IT ALISE” dalībnieku kolektīvā, bieži tiek rīkotas SCRUM sanāksmes. Sanāksmēs tiek apspriesti problēmu pieteikumi, termiņi, grūtības ar kurām nākas sastapties, veicot uzdevumus vai risinot pieteikumus un izvirzīti iespējamie risinājumi grūtību pārvarēšanai. Veicot „BIS ALISE” portāla izstrādi, šādās sanāksmēs tika izvirzītas sistēmas prasības, novērtēta esoša funkcionalitāte un izteikti iespējamie uzlabojumi. Sanāksmēs arī tika panākta vienotība par sistēmas izskatu, logotipu un funkcionalitātes izvietojumu noteiktās portāla sadaļās.

Ik pēc divām nedēļām tiek rīkotas sanāksmes, kurās tiek apspriesta turpmāka vadības stratēģija, pamatojoties uz darbinieku viedokļiem. Šādās sanāksmēs tiek izteiktas lietas, kuras ir patikušas un nav patikušas, secīgi tiek izvirzīti priekšlikumi šo lietu uzlabošanai.

2. IZSTRĀDĀTAS SISTĒMAS RAKSTUROJUMS

Prakses gaitā tika izstrādāts „BIS ALISE” sistēmas klientu informatīvais portāls. Šajā nodaļā tiek aprakstīts uzdevumu risināšanas process, ar to saistītie pētījumi, izmantotas tehnoloģijas un iegūtie rezultāti.

2.1. Risināmie uzdevumi un sistēmas apraksts

Prakses ietvaros autoram tika uzticēti sekojošie uzdevumi:

- Izstrādāt bibliotēku informācijas sistēmas „ALISE” klientu informatīvo portālu;
- Veikt sistēmas izstrādi .NET vidē, izmantojot SQL Server datu bāzi;
- Paredzēt sistēmas modularitāti, balstoties uz sistēmas projektēšanas šabloniem un paņēmieniem;
- Veikt klientu bāzes pārveidi jaunajā sistēmā;
- Izstrādāt aizsargāto klientu sadaļu;
- Paredzēt iespēju koriģēt tīmekļa lapu satura tekstu, pievienot dokumentus, attēlus un cita veida datus;
- Izstrādāt sistēmas administratīvo moduli;
- Veikt sistēmas ieviešanu un lietotāju apmācību.

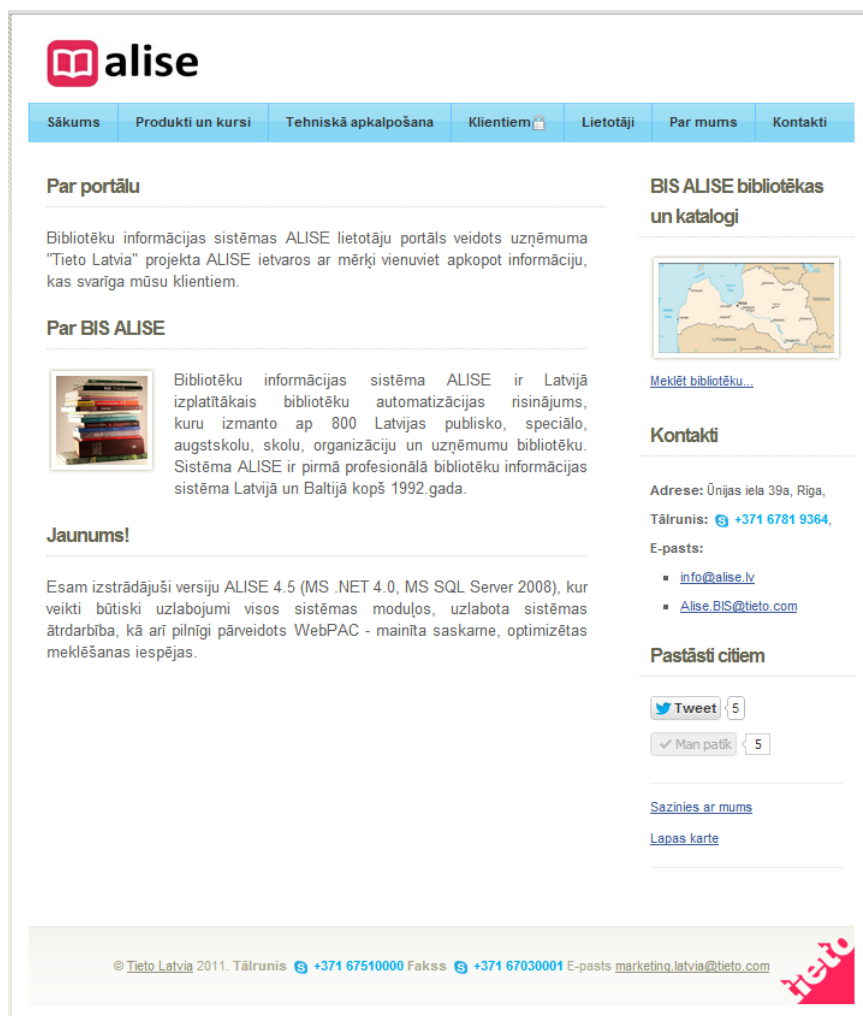
Veicot uzdevumu izpildi, tika iegūts Web lietojums, kurā ir izklāstīta informācija par „BIS ALISE” sistēmas produktiem, „IT ALISE” komandas piedāvātiem pakalpojumiem, tehnisko atbalstu un rīkotiem lietotāju apmācības kursiem. Ir sniegts uzņēmuma „Tieto Latvia” apraksts un kontaktinformācija, ir minētas tā kompetences un darbības jomas, kā arī ir paredzēts ieskaits bijušā „IT ALISE” uzņēmuma attīstības vēsturē.

Portāls satur aizsargāto sistēmas klientu sadaļu, kurā ir izvietota lietotāju dokumentācija, metodiskie un mācību materiāli saistībā ar pēdējām „BIS ALISE” sistēmu versijām. Šajā sadaļā ir pieejama notikušo semināru apraksts un fotogalerijas, kā arī ir piedāvāta darbam ar sistēmām nepieciešamo instalācijas datņu lejupielādēšanas iespēja.

Portāls piedāvā lietotājiem iespēju sazināties ar atbalsta servisu, „IT ALISE” komandu, meklēt bibliotēkas pēc vairākiem kritērijiem un ienākt klientu mājas lapā, kas ir viens no sistēmas „ALISE” produktiem – „WebPAC”.

Sistēma paredz administratīvo moduli, kurā ir iespējams rediģēt tīmekļa lapu saturu, pievienot jaunas un dzēst esošas tīmekļa lapas, augšupielādēt dokumentus, attēlus u.t.t., kā arī labot, pievienot un dzēst sistēmas vadības datus, sūtīt ziņojumus klientiem u.tml.

„BIS ALISE” lietotāju portāls ir attēlots 2.1. attēlā.



2.1. att. Portāla „BIS ALISE” sākumlapa

2.2. Sistēmas arhitektūras projektēšana

Uzsākot sistēmas izstrādi, sākotnēji tika risināti sistēmas arhitektūras projektēšanas jautājumi. Sistēmas modularitātes nodrošināšanai, tika nolemts izmantot MVC projektēšanas paņēmieni. Paņēmienu ideja paredz lietotāja saskarnes nodalīšanu no lietojumprogrammas biznesa modeļa [8]. Tas tiek panākts ar triju klašu vienību palīdzību:

- Modelis – lietojumprogrammas biznesa modeļa datu vienības objekts, kurš iekapsulē noteikto stāvokli un uzvedību atbilstoši kādai sistēmas funkcijai.

Bieži vien modelis reprezentē noteiktas saskarnes (tīmekļa lapas) loģiku un datus. Objekts maina savu stāvokli attiecībā pret veiktām darbībām kontrolierī vai saskarnē un ziņo par izmaiņām vai nu kontrolierim, ja izmaiņas ir veiktas ar lietotāja starpību, vai nu skatam vizuālo datu atjaunošanai, ja izmaiņas ir veiktas kontrolierī.

- Skats – vienība, kas paredz vizuālo datu attēlošanu;
- Kontrolieris – apstrādā lietotāju pieprasījumus, veicot kādas noteiktas funkcijas, kuras ietekme modeļa stāvokli.

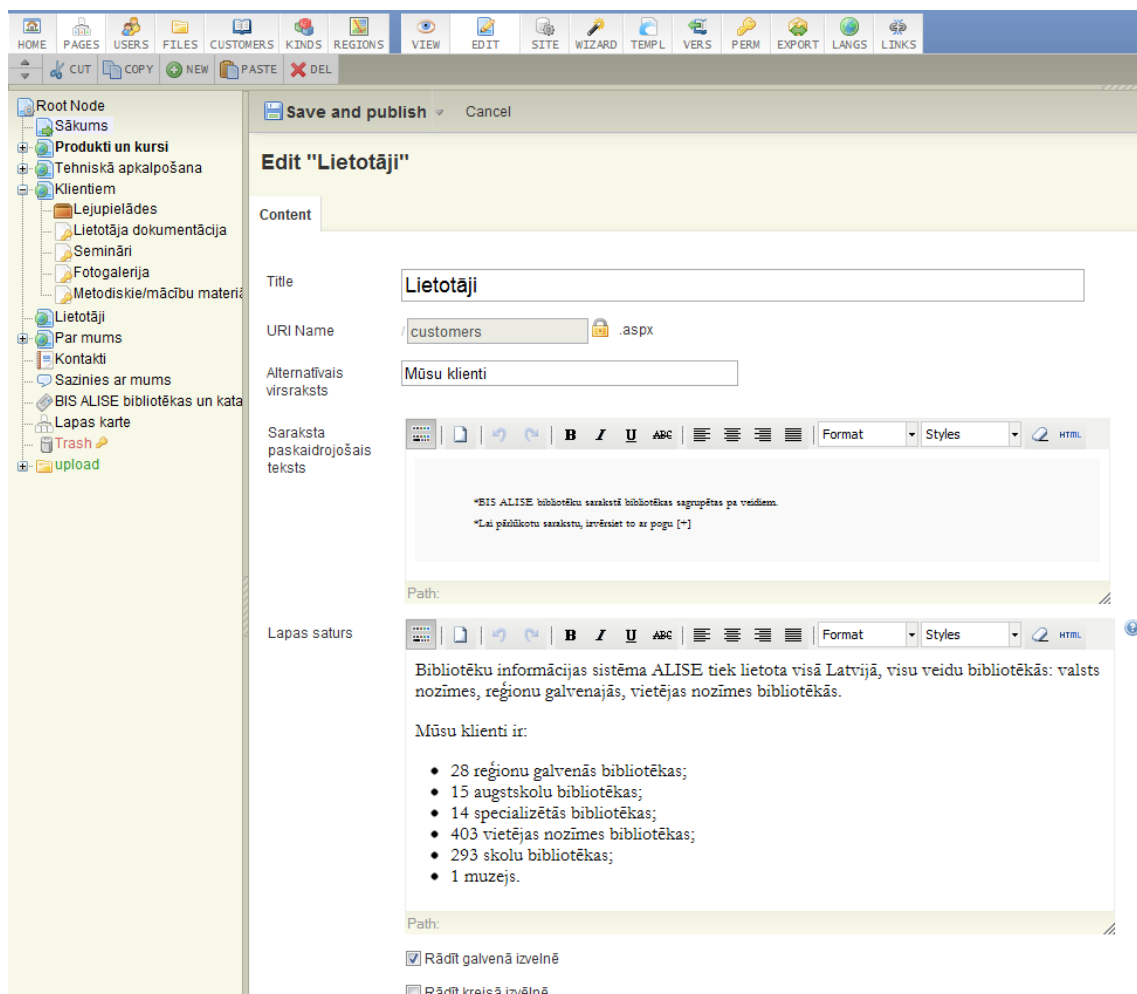
Balstoties uz pieņemto lēmumu tika izvēlēta ASP.NET MVC tehnoloģija. Par sistēmas valodu tika izvēlēta C# programmēšanas valoda.

Lai nodrošinātu iespējas lietotājiem (kolēģiem) rediģēt tīmekļa lapu satura tekstu, veidot jaunas nodaļas un tīmekļa lapas, pievienot klāt attēlus, dokumentus un instalācijas datnes, kā arī administrēt un pārvaldīt sistēmas datus tika izvēlēts N2 CMS ietvars. N2 CMS ir atvērta satura vadības sistēma, kas uz doto brīdi ir savietojama ar ASP.NET MVC un ASP.NET WebForms tehnoloģijām. Sistēmas būtība ir piedāvāt Web lietojuma vadības mehānismu, paredzot sistēmas struktūras, piemēram, tīmekļa lapu un to saturu, glabāšanu datu bāzē. Šādām struktūrām tiek piesaistītas veidnes, kuras pēc būtības reprezentē kādu noteiktu ierakstu CMS datu bāzes tabulā. Veidne ietver noteiktos laukus, kuri atbilst glabājamiem ieraksta laukiem datu bāzes tabulā. Struktūrvienību izveidošanas, dzēšanas un modificēšanas funkcijas tiek veiktas speciāla lietojuma vadības administrācijas modelī. Portāla „BIS ALISE” N2 CMS administrācijas modulis ir attēlots 2.2. attēlā.

2.2. attēlā ir attēlota tīmekļa lapas „Lietotāji” veidne, kurai ir raksturīgi tādi lauki, ka „lapas saturs”, „virsraksta paskaidrojošais teksts”, „radīt labajā izvēlnē” u.t.t.

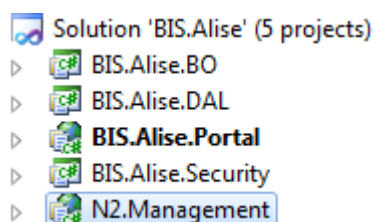
N2 CMS ietvars nodrošina programmatūras interfeisu kopumu, ar kuru palīdzību ir veikta lietojuma struktūras izstrāde. Lai pielāgotu N2 CMS administrācijas modelī portāla vajadzībām, t.i., nodrošinātu specializētas administratīvas funkcijas saistība ar sistēmas vadāmiem datiem, ir lejupielādēts dota modeļa pirmteksts un papildināts ar praktikanta pirmtekstu doto funkciju realizēšanai.

Sistēmas projektēšanas posmā tika nolemts glabāt sistēmas vadības datus atsevišķā datu bāzē, šādi izdalot atsevišķo datu bāzi sistēmas datiem un portāla struktūrai. Saskaņā ar izvēlēto MVC projektēšanas principu un N2 CMS ietvars arhitektūru, tika nolemts projektēt portāla sistēmas arhitektūru balstoties uz „trīs līmeņa arhitektūras” (three-tier architecture) principu.



2.2. att. Portāla „BIS ALISE” administrācijas modulis

Princips paredz sistēmas lietotāja saskarnes, datu pieejas līmeņa un biznesa loģikas atdalīšanu atsevišķos moduļos [9]. Sistēmas projekta struktūra ir uzskaitīta 2.3. attēlā.



2.3. att. Sistēmas projekta struktūra.

Attēloto sistēmas apakšprojektu (moduļu) nozīme ir sekojoša:

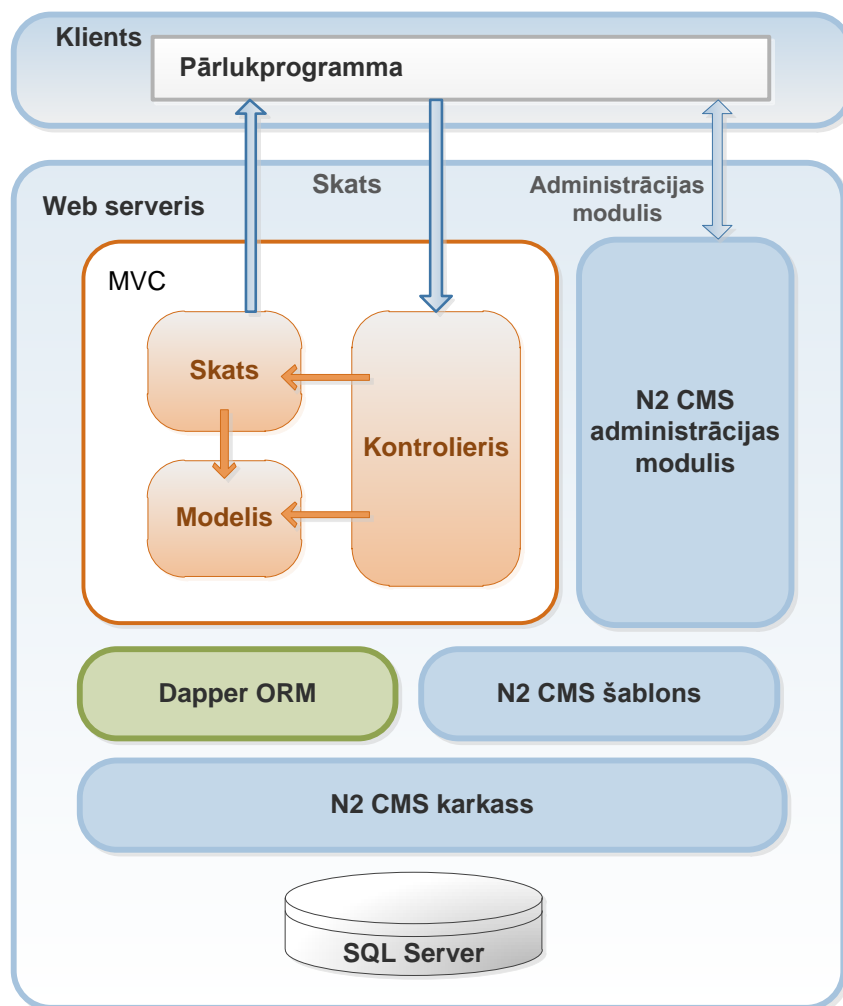
- „Bis.Alise.BO” – ietver sevī lietojuma domēna jeb biznesa objektus;
- „Bis.Alise.DAL” – datu pieejas līmenis, kas nodrošina darbu ar datu bāzi;
- „Bis.Alise.Portal” – lietotāju saskarnes modulis. Satur pēc MVC principa izstrādātus modeļus, kontrolierus un skatus. Ietver atsevišķo loģiku saskarnes

funkcionalitātes nodrošināšanai, kā arī N2 CMS komponentus un izstrādātos servisu;

- „Bis.Alise.Security” – paredz funkcionalitāti lietotāju kontu un lomu vadībai, paroli šifrēšanai, ģenerēšanai u.t.t.;
- „N2.Management” – pārstrādātais N2 CMS administrācijas modulis.

Datu pieejas līmeņi nolēmu projektēt pēc „repozitorija šablona” (*repository pattern*) principa. Šablons nosaka noteikta veida disciplīnu datu pieejas līmeņa implementēšanai, kas ļauj uzlabot pirmteksta lasāmību, novēršot iespējamo pirmteksta dublēšanu, paredz centralizētu datu pieejas mehānismu un nodrošina pieeju koda labākai testēšanai ar vienības testu palīdzību [10]. Darbam ar datu bāzi tika nolemts izmantot Dapper objekt-relāciju kartēšanas, turpmāk ORM, ietvaru.

Izstrādāta sistēmas arhitektūra ir parādīta 2.4. attēlā.



2.4. att. Portāla „BIS ALISE” arhitektūra

2.3. Uzdevumu risināšana

Sistēmas izstrādē tika ieguldīts pamatīgs darba apjoms. Izstrādes gaitā tika noteiktas jaunas sistēmas funkcijas, kā arī pārstrādāti daži no esošiem risinājumiem. Rezultātā tika iegūta pilnvērtīga sistēma, kas pilda savas funkcijas attiecībā pret nostādītām prasībām. Dotajā sadaļā ir konspektīvi aprakstīti uzdoto uzdevumu risināšanas paņēmieni, abstrahējoties no detaļām un papildus funkcionalitātes izstrādes apraksta.

2.3.1. Lietojuma struktūras izstrāde

Sistēmas struktūra ir izstrādāta balstoties uz MVC modeļa ideoloģijas principu, pielietojot ASP.NET MVC tehnoloģijas un N2 CMS līdzekļus.

Sistēmas izstrādes process sākas ar N2 CMS bāzes veidnes izstrādi, izmantojot OOP mantošanas principu. Bāzes veidnes pirmteksts ir attēlots 2.5. attēlā.

```
01 [WithEditableTitle("Virsraksts", 4, ContainerName = Tabs.Content)]
02 [TabContainer(Tabs.Content, "Content", 0)]
03 public abstract class BasePage : ContentItem {
04
05     [EditableText("Alternatīvais virsraksts", 90,
06         HelpTitle = "Lapas alternatīvais virsraksts")]
07     public virtual string PageTitle { get; set; }
08
09     [EditableFreeTextArea(
10         "Lapas saturs", 100,
11         ContainerName = Tabs.Content,
12         HelpTitle = "Lapas satura teksts")]
13     public virtual string Text {
14         get { return (string) (GetDetail("Text") ?? string.Empty); }
15         set { SetDetail("Text", value, string.Empty); }
16     }
17 }
```

2.5. att. Bāzes veidnes pirmteksts

Bāzes veidnes klase satur kopīgos atribūtus, kuri tiks mantoti pārējās veidņu klases. Pēc attēla ir redzams, ka klase satur divus laukus: „Alternatīvais virsraksts” un „Lapas saturs”, virs kuriem ir definēti kāda noteikta tipa saskarnes kontroļu atribūti. Dota piemēra šādas kontroles ir parastais un izvēršamais teksta lauks (skat. 2.2. attēlā attēloto „Lietotāji” tīmekļa lapu rediģēšanas režīmā). Dotiem atribūtiem piemīt parametri, kuri nosaka lauka paskaidrojošo tekstu, izvietojumu u.t.t., līdzīgi ir ar atribūtiem, kuri raksturo pašu klasi.

Tika izstrādāta arī sadaļas tīmekļa lapu bāzes klašu veidnes, kuras bez mantotiem atribūtiem definē arī katrai sadaļai kopīgos atribūtus. Šāds sadalījums ļauj atlasīt noteikta tipa objektus, piemēram, portālā izvēlnes sastādīšanai (skat. 2.1 un 2.2 attēlus, izvērsto tīmekļa lapu sarakstu un lietojuma galveno izvēlni).

Nākamā solī ir iespējams veikt portāla struktūras izveidi. Vispirms ir nepieciešams izstrādāt modeli. Bieži vien modelis ir pati tīmekļa lapas veidne, jo vairākos gadījumos ir nepieciešams tikai izvadīt tīmekļa lapas saturu, kas glabājas N2 CMS datu bāzē. Šāda tipa modelis ir attēlots 2.6. attēlā.

```

01 [PageDefinition(„Lietotāju lapa“,
02     Description = "Izveidot jaunu lietotāju lapu.",
03     IconUrl = "~/N2/Resources/icons/page_world.png")]
04 [RestrictParents(typeof (IRootPage))]
05 [WithEditableTitle, WithEditableName]
06 [RestrictCardinality(MaximumCount = 1)]
07 public class UsersPage : BaseChapter, IStructuralPage {
08
09     [EditableFreeTextArea("Saraksta paskaidrojošais teksts",
10         100, Required = true,
11         RequiredMessage = "Paskaidrojošais teksts nav norādīts",
12         HelpTitle = "Klientu saraksta paskaidrojošais teksts")]
13     public virtual string Description { get; set; }
14 }

```

2.6. att. Sadaļas tīmekļa lapas „Lietotāji” veidnes klase

Atsevišķos gadījumos, piemēram, kad papildus tīmekļa lapas satura datiem ir nepieciešams noteiktā veidā izvadīt arī sistēmas vadības datus, ir iespējams izstrādāt specializēto modeli. Šāda tipa modelis ir parādīts 2.7. attēlā, kurā papildus „FindLibraryPage” veidnes instancei tiek ietverti arī cita tipa objekti. Bieži vien modelī definēto mainīgo nozīme ir glabāt lietotāja ievadīto vērtību, kas tiks nodota kontrolierim.

```

01 public class SearchModel : IItemContainer<FindLibraryPage> {
02     public string Title { get; set; }
03     public int RegionId { get; set; }
04     public IPagedList<Customer> PagedCustomers { get; set; }
05     public FindLibraryPage CurrentItem { get; set; }
06     ContentItem IItemContainer.CurrentItem {
07         get { return CurrentItem; }
08     }
09     ...
10 }

```

2.7. att. Bibliotēku meklēšanas tīmekļa lapas modeļa fragments

Implementējot kontrolieri, ir nepieciešams norādīt modeli, ar kuru tas darbosies. Kontroliera iespējama implementācija ir attēlota 2.8. attēlā.

```

01 [Controls(typeof(Users))]
02 public class UsersController : ContentController<UsersPage> {
03
04     public override ActionResult Index() {
05         return View(CurrentItem);
06     }
07
08     [HttpPost]
09     public JsonResult GetUsers(string root) {
10         var nodes = root == "source" ?
11             GetRoots() : GetChildren(int.Parse(root));
12         return Json(nodes);
13     }
14
15     public IEnumerable<JsonTreeNode> GetChildren(int kindId) {
16         var connectionString = SessionManager.Get(
17             SessionKeys.CnStringKey, HttpContext,
18             () => WebConfigurationManager.
19                 ConnectionStrings["BisAlise"].
20                 ConnectionString);
21
22         var repository = new CustomerRepository(connectionString);
23         var items = repository.Filter(kindId);
24
25         return items.Select(item => new JsonTreeNode {
26             id = item.Id.ToString(),
27             text = GetNodeDisplayText(item),
28             hasChildren = false
29         }).ToList();
30     }
31     ...

```

2.8. att. Tīmekļa lapas „Lietotāji” kontroliera realizācijas fragments

Kontrolieris ir atbildīgs par tīmekļa lapas funkcijām, secīgi visa tīmekļa lapas loģika tiek implementēta tieši šeit. Lietotājam pieprasot dota resursu, Web serveris izpildīs kontroliera „Index” metodi, kura atgriezīs lietotāja saskarni jeb skatu, iepriekš nododot tām modeli. Šāds pieprasījumu apstrādes princips balstās uz saņemtas URL adreses apstrādi, kuras rezultātā tiks konstatēts izpildāmais kontrolieris un metode, kuru būtu jāizpilda.

Kontrolieris var ietvert sevī vairākas šāda tipa metodes, kuru izpildes rezultātā tiks atgriezts skats. Lai norādītu kādu noteiktu metodi, kurai ir jāizpildās kādas atsevišķas darbības rezultātā, piemēram, izvadīt nākamās ierakstus, lietotājam pārslēdzoties starp lappusēm

(*paging*), tai ir jādefinē „HttpPost” atribūts. Citādi jauno pieprasījumu gadījumu vienmēr tiks izsaukta metode pēc noklusējuma – „Index”. 2.8. attēlā parādīta kontrolierī šādā metode ir „GetUsers”, kura paredz izvēršama lietotāja saraksta izvadi. Lai nodrošinātu asinhrono datu izvades mehānismu, t.i., izvadīt datus tikai tad, kad tie tiek reāli pieprasīti nevis ielādēt visu sarakstu uzreiz, metodei ir jāizpildās katra saraksta izvēršanas pieprasījuma gadījumā.

Metode savos parametros var saņemt jebkāda tipa vērtību atkarībā no sagaidāma rezultāta. Piemēram, var saņemt modeli, reaģēt uz tā izmaiņām, mainīt tā stāvokli un atkārtoti nodod skatam.

Izstrādājot tīmekļa lapas modeli un kontrolieri, ir jāizstrādā arī skats. Skats ir tīmekļa lapas vizuālais elements, kas definē tas izskatu ar Html elementu palīdzību. Skats var saturēt elementus formas pārraidīšanai (submit), kuras rezultātā tiks mainīts modeļa stāvoklis. Modelis tiek izmantots datu attēlošanai, piemēram, 2.9. attēlā parādītajā skatā dotam mērķim tiek izmantoti „Title” un „PageTitle” lauki.

```

01 <%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
02     Inherits="ViewPage<UsersPage>" %>
03
04 <asp:Content ID="HC" ContentPlaceHolderID="TC" runat="server">
05     <%= Html.Display("Title") %>
06 </asp:Content>
07 <asp:Content ID="BC" ContentPlaceHolderID="MC" runat="server">
08     <script src="../../Content/Scripts/jquery-treeview-async-0.1.0.js"
09         type="text/javascript" />
10     <script type="text/javascript">
11         $(document).ready(function () {
12             $('#customers').treeview({
13                 url: '/Users/GetUsers/',
14                 async: true,
15                 collapsed: true,
16                 ajax: {
17                     type: 'post'
18                 }
19             });
20         });
21 </script>
22 <div class="post">
23     <div class="header">
24         <h3><%=Html.Display("PageTitle") %></h3>
25     </div>
26     <div class="content">
27         <%=Html.DisplayContent("Description") %>
28         <br />
29         <div><ul id="customers" class="treeview-fam"></ul></div>
30     </div>
31 </div>
32 </asp:Content>

```

2.9. att. Tīmekļa lapas „Lietotāji” skata fragments

2.3.2. Sistēmas domēna un datu pieejas līmeņa izstrāde

Veicot sistēmas izstrādi viens no svarīgiem uzdevumiem bija reģistrēt visus sistēmas „ALISE” lietotājus, sastādot pilno klientu bāzi. Reģistrētiem lietotājiem bija jāparedz iespēja autorizēties sistēmas aizsargātājā sadaļā, kā arī jānodrošina paroles nomaiņas un atjaunošanas funkcijas. Sistēmai bija jāparedz arī administratīvai modulis, lietotāju kontu un sistēmas datu pārvaldībai.

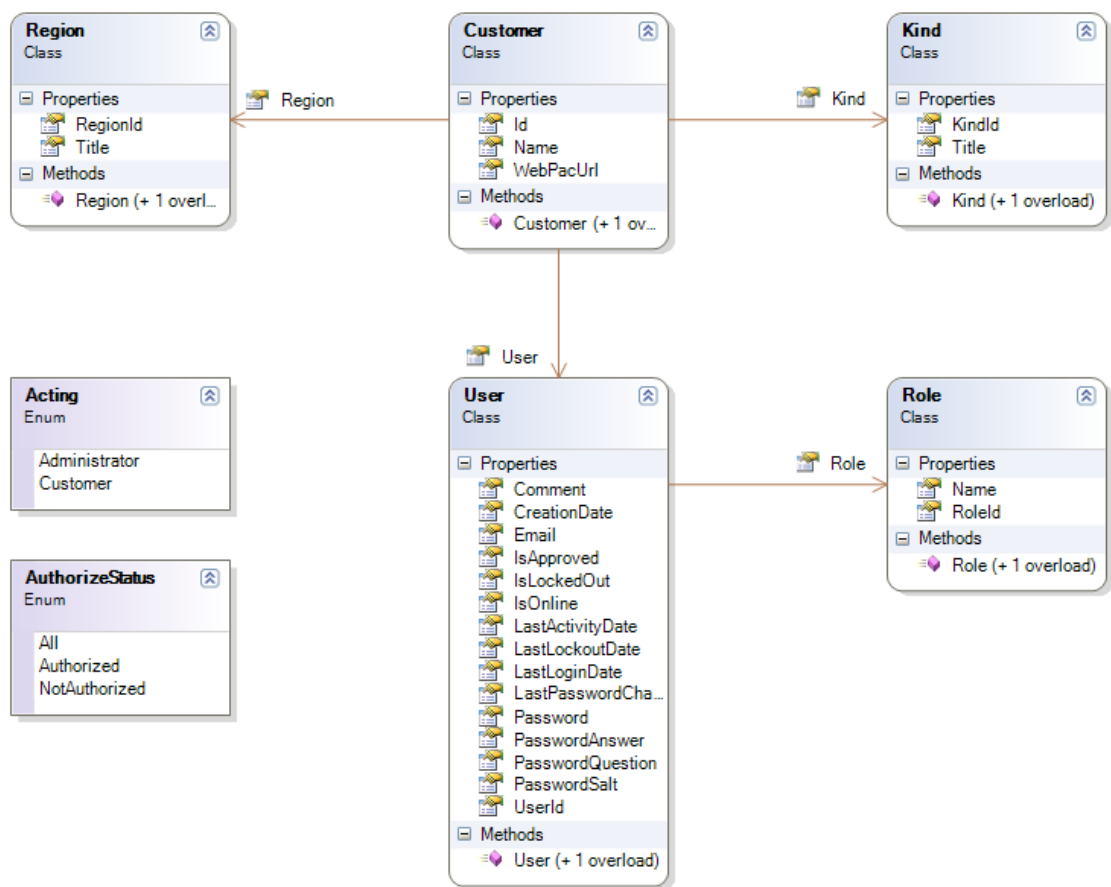
Balstoties uz dotām prasībām, tika izstrādātas sekojošas entītijas jeb objekti:

- Klients – datu vienību, kas nosaka no sistēmas viedokļa nozīmīgos bibliotēku datus.
- Lietotāji – nosaka lietotāja funkcijas, saturot laukus pēdējas aktivitātes uzskaitē, e-pastu, aizsargāto paroļu u.c. datu glabāšanai.
- Lomas – definē sistēmas lietotāju lomas, nodrošinot atbilstošas pieejas tiesības. Sistēmā pastāv divi lietotāju tipi: administrators un klients. Vienīgi administratoriem ir iespējams pieslēgties administrācijas moduļim.
- Novadi – satur datus par novadiem. Objekts tika realizēts bibliotēku atlases funkcijas nodrošināšanai, izvēlētos kādu noteikto novadu.
- Veidi – pakārto bibliotēkas atbilstoši noteiktiem bibliotēku veidiem. Objekts tika paredzēts bibliotēku atlases funkciju nodrošināšanai, kā arī izvēršama klientu saraksta izstrādei, ļaujot lietotājiem pretī atbilstošam veidam redzēt piederošas bibliotēkas.

Izstrādātais sistēmas biznesa domēna modelis ir parādīts 2.10. attēlā.

Objektu datu glabāšanai tika izstrādāta atsevišķa datu bāze. Izstrādājot datu pieejas līmeni, tika paredzēti vairāki bāzes interfeisi un klases, kopīgo un katrai atsevišķai entītijai raksturīgo datu pieejas operāciju izdalīšanai. Šāda pieeja uzlabo pirmteksta atklūdošanas efektivitāti, nodrošinot iespēju izstrādāt centralizētus testus datu pieejas operāciju testēšanai, kā arī ļauj noteiktas klases ietvaros, definējot bāzes klases vai interfeisa atsauci, atsaucoties uz mantotas klases objektu. 2.11. attēlā ir parādīts bāzes interfeiss, kopīgo operāciju definēšanai.

Darbam ar datu bāzi ir izmantots Dapper ORM ietvars. Dapper ir salīdzinoši viegls savas kategorijas rīks, salīdzinot ar Hibernate (NHibernate), ADO.NET Entity Framework, JPA u.c. ORM tehnoloģijām. Dapper priekšrocības ir augsta veiktspēja un viegla lietošana, kura neprasa papildus literatūras avotu izlasīšanu, piemēram, veiktspējas optimizācijai, kas bieži vien ir raksturīgs citām ORM tehnoloģijām [11].



2.10. att. Sistēmas biznesa domēna modelis

Dapper ORM ietvars paredz ADO.NET tehnoloģijas uzlabošanu, piedāvājot līdzekļus izgūto ierakstu kartēšanai lietojuma domēna entītijās. Dotais risinājums labi der projektos, kuros biznesa domēna modelis ir samērā vienkāršs – neiekļauj sarežģītus biznesa koncepcijas. Dapper lietošanas piemērs ir parādīts 2.12. attēlā.

```

01 public interface IRepository<T> {
02     void Create(T t);
03     void Update(T t);
04     void Delete(int id);
05     T GetById(int id);
06     IEnumerable<T> GetAll();
07 }

```

2.11. att. Datu pieejas līmeņa bāzes interfeiss

```

01 public class RoleRepository : Repository<Role>, IRoleRepository {
02
03     public override Role GetById(int roleId) {
04         using (var conn = new SqlConnection(ConnectionString)) {
05             conn.Open();
06             return conn.Query<Role>(
07                 "select * from Roles where RoleId = @id",
08                 new {id = roleId}).FirstOrDefault();
09         }
10     }
11 }

```

2.12. att. Dapper ORM pielietošanas piemērs

2.3.3. Aizsargātas klientu sadaļas izstrāde

Klientu reģistrēšana sistēmā norisinājās ar atsevišķas programmas izstrādi, kura veica saņemta Excel dokumenta ielasīšanu un ielasīto datu pievienošanu datu bāzē. Dotais Excel dokuments saturēja datus par klientiem, novadiem un veidiem, kā arī parādīja savstarpējas saistības starp datiem, t.i., kādam novadam un veidam pieder bibliotēka.

Nākamā izstrādes posmā bija nepieciešams izstrādāt lietotāju funkcijas, kas paredzētu tādas darbības, ka autorizācijas datu pārbaude, paroles maiņa un atjaunošana, jauna lietotāja pievienošana, dzēšana un tā datu labošana u.t.t. Dotam nolūkam ir implementēta ASP.NET tehnoloģijas MembershipProvider bāzes klase, kura definē kopējas lietotāju kontu pārvaldībai nepieciešamas funkcijas.

Klases implementācija ir veikta ar nolūku paredzēt specifisku loģiku lietotāju veicamais funkcijās, piemēram, autorizācijas veikšana ar e-pasta adreses nevis lietotājvārda palīdzību. Klases implementācijas fragments ir attēlots 2.13. attēlā.

```

01 public class AliseMembershipProvider : MembershipProvider {
02
03     public override bool ChangePassword(string username,
04         string oldPassword, string newPassword) {
05
06         User u = _repository.GetUserByEmail(username);
07         if (u == null || u.Password != ToStoredPassword(oldPassword,
08             u.PasswordSalt))
09             return false;
10         u.PasswordSalt = CryptographyProvider.CreateSalt();
11         u.Password = ToStoredPassword(newPassword, u.PasswordSalt);
12         _repository.Update(u);
13         return true;
14     }
15 }

```

```

16     public override bool ValidateUser(string username,
17                                     string password) {
18         var u = _repository.GetUserByEmail(username);
19         if (u != null && u.Password == ToStoredPassword(password,
20             u.PasswordSalt)) {
21             u.LastLoginDate = DateTime.Now;
22             u.LastActivityDate = u.LastLoginDate;
23             _repository.Update(u);
24             return true;
25         }
26         return false;
27     }
28
29     public override int GetNumberOfUsersOnline() {
30         var users = _repository.GetAll();
31         if (users == null)
32             return 0;
33         int userIsOnlineTimeWindow = (
34             Membership.UserIsOnlineTimeWindow > 0 ?
35             Membership.UserIsOnlineTimeWindow : 20);
36         return users.Count(u => u.LastActivityDate > DateTime.Now
37             .AddMinutes(-userIsOnlineTimeWindow));
38     }
39     ...

```

2.13. att. MembershipProvider klases implementācijas fragments

Līdzīgi tika implementēta RoleProvider ASP.NET bāzes klase, lai realizētu lomu pārvaldības funkcijas. Šādas funkcijas ir lomu izveidošana, izgūšana, pārbaude vai lietotājs ietilpst uzdotajā lomā, lietotāju pievienošana un dzēšana no definētam lomām u.c. ar lomam saistītas darbības.

Balstoties uz izstrādātu funkcionalitāti tika izveidotas lietotāju autorizācijas, paroles maiņas un atjaunošanas tīmekļa lapas. Attiecīgi kontrolieros izsaucot implementēto klašu metodes, lai veiktu lietotāju pieprasījumu apstrādi.

Problēmas rādās ar aizsardzības mehānisma realizēšanu, kurā lietotājs var piekļūt aizsargātam resursam noradot tiešo URL adresi, šādi izvairoties no autorizācijas. ASP.NET paredz iekšējos līdzekļus šādas problēmas novēršanai, definējot atbilstošos iestatījumus ASP.NET lietojuma konfigurācijas Web.config failā. Par cik izstrādājamā portālā tika paredzētas divas autorizācijas tīmekļa lapas: N2 CMS administrācijas moduļa un klientu aizsargātas sadaļas piekļūšanai, šādi iestatījumi tika jau aizņemti integrējot N2 CMS sistēmas līdzekļus. Aizsardzības iestatījumi konfigurācijas failā N2 CMS autorizācijas tīmekļa lapai ir parādīti 2.14. attēlā.

01	<authentication mode="Forms">
02	<forms loginUrl="n2/login.aspx" protection="All" timeout="30000"
03	path="/" />
04	</authentication>

2.14. att. Aizsardzības iestatījumi

Problēmas novēršanai tika izstrādāts patstāvīgs risinājums izmantojot ASP.NET MVC tehnoloģijas līdzekļus, kuri tika pielietoti lietotāja autorizācijas statusa, ienākoša pieprasījuma URL adreses noteikšanai un pāradresēšanai uz autorizācijas tīmekļa lapu gadījumā, ja lietotājs nav autentificēts. Izstrādātais risinājums ir klases atribūts, kuru ir jādefinē visiem kontrolierim, kas satur aizsargāto saturu. Risinājuma realizācija ir parādīta 2.15. attēlā.

01	public class SignInAttribute : ActionFilterAttribute {
02	public override void OnActionExecuting(
03	ActionExecutingContext filterContext) {
04	
05	if (!filterContext.HttpContext.User.Identity.IsAuthenticated) {
06	var u = filterContext.HttpContext.Server.UrlEncode(
07	filterContext.HttpContext.Request.RawUrl);
08	var helper = new UrlHelper(filterContext.Controller
09	.ControllerContext.RequestContext);
10	var url = helper.Action("LogOn", "Account",
11	new { returnUrl = u });
12	filterContext.Result = new RedirectResult(url);
13	}
14	base.OnActionExecuting(filterContext);
15	}
16	}

2.15. att. Klientu aizsargātas sadaļas aizsardzības mehānisma realizēšana

Izstrādājot aizsargāto klientu sadaļu un nodrošinot aizsardzības mehānismu patvaļīgas piekļūšanas iespējas novēršanai aizsargātiem resursiem, bija arī jāmodificē portālā lapu kartes izvades funkcionalitāte. Bija jāpanāk, lai aizsargātas tīmekļa lapas radītos tikai tad, kad klients ir autorizējies sistēmā. Par cik sistēmas struktūra balstās uz N2 CMS veidņu realizāciju, tika izstrādāts tīmekļa lapu filtrs, izmantojot N2 CMS līdzekļus. Filtra klases realizācijas fragments ir attēlots 2.16. attēlā.

01	public class SiteMapFilter : ItemFilter {
02	private readonly IPrincipal _user;
03	
04	public override bool Match(ContentItem item) {
05	if (!item.IsPage !item.IsPublished() item.IsExpired()
06	!item.Visible)
	return false;

```

07         return !(item is ISecurablePage) ||
08                ((ISecurablePage)item).IsAuthorizedToView(_user);
09     }
10     ...
}

```

2.16. att. Izstrādāta tīmekļa lapu filtra fragments

Attēlā ir redzams, ka izgūstamas tīmekļa lapas atbilstība tiek noteikta pēc vairākiem kritērijiem, piemēram, vai tīmekļa lapa ir redzama, publicēta un joprojām ir aktīva. Šeit arī tiek pārbaudīts vai tīmekļa lapa ir aizsargāta un vai lietotājs ir autorizēts to redzēt. Šīs mehānisms balstās uz izstrādāto N2 CMS veidni un interfeisu, kuru koplieto visas aizsargātas tīmekļa lapas. Veidnes realizācijas pirmteksts ir attēlots 2.17. attēlā.

```

01 [PageDefinition("Aizsargāta satura lapa",
02     Description = "Izveidot jaunu aizsargāto satura lapu",
03     IconUrl = "~/N2/Resources/icons/page_white_key.png")]
04 [RestrictParents(typeof(IStructuralPage))]
05 [WithEditableTitle, WithEditableName]
06 public class SecurablePage : BasePage, ISecurablePage {
07     public bool IsAuthorizedToView(IPrincipal user) {
08         return user.Identity.IsAuthenticated &&
09                (user.IsInRole(Acting.Administrator.GetStringValue()) ||
10                 user.IsInRole(Acting.Customer.GetStringValue()));
11     }
12 }
13 }

```

2.17. att. Aizsargātas tīmekļa lapas veidne

Pēdējais posms paredzēja „robots.txt” datnes izstrādi, kas aizliedza Google, Yandex u.c. kompāniju robotiem indeksēt aizsargātos portālā resursus. Datnes saturs ir attēlots 2.18. attēlā.

```

01 User-agent: *
02 Disallow: /upload/
03 Disallow: /Views/Account/
04 Disallow: /Views/Customer/
05 Disallow: /Views/Tools/
06 Disallow: /Views/ForCustomer/

```

2.18. att. „Robots.txt” datnes saturs

2.3.4. Administrācijas moduļa izstrāde

Sistēmas izstrādes ietvaros bija nepieciešams izstrādāt administrācijas moduli sistēmas datu vadībai. Bija jāparedz šādas iespējas:

- Labot, dzēst un pievienot novadus;
- Labot dzēst un pievienot veidus;
- Labot, dzēst un reģistrēt jaunus klientus, nosūtot viņiem e-pasta ziņojumus par autorizācijas iespēju. Bija jāparedz arī tādas funkcijas, ka e-pasta ziņojumu nosūtīšana izvēlētiem klientiem, klientu meklēšana un atlasīšana pēc vairākiem kritērijiem, kā arī iespēja labot e-pasta ziņojuma teksta veidnes;
- Labot dzēst un reģistrēt jaunus lietotājus, nosakot viņu lomas. Bija jāparedz tādas funkcijas, ka paroli maina, brīdinot lietotājus par izmaiņām, lietotāju meklēšana pēc atslēgvārdiem, kā arī redzēt lietotājus, kuri pašlaik atrodas portālā.

Prasību realizēšanai tika nolemts pārstrādāt N2 CMS administrācijas moduli, piedāvājot iespēju vienuviet gan uzturēt portāla struktūru, gan pārvaldīt sistēmas datus. Uzdevuma veikšanai tika leģupielādēts administrācijas modeļa pirmteksts, kurā tika pārstrādāta esošo lietotāju sadaļa un izstrādātas jaunas klientu, novadu un veidu administrācijas sadaļas. Modulis ir attēlots 2.19. attēlā, kurā ir atvērta klientu sadaļa.

Bibliotēkas nosaukums	E-pasts	Novads	Veids	WebPac
Ādažu bibliotēka	mirdza@adazi.lv	Ādažu novads	Vietējas nozīmes bibliotēkas	http://salaspils.biblioteka.lv/Alise/v25/home.aspx dzēst
Āgenskalna Valsts ģimnāzijas bibliotēka	avj@riga.lv	Rīgas pilsēta	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://biblioteka.rimc.lv/Alise/v20/home.aspx dzēst
Aglonas novada centrālā bibliotēka	biblioteka@aglonas.lv	Aglonas novads	Vietējas nozīmes bibliotēkas	http://preili.biblioteka.lv/Alise/v4/home.aspx dzēst
Aglonas vidusskolas bibliotēka	ilzeg5@inbox.lv	Aglonas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://preili.biblioteka.lv/Alise/v22/home.aspx dzēst
Ainažu pilsētas bibliotēka	rutab-ka@inbox.lv	Salacgrīvas novads	Vietējas nozīmes bibliotēkas	http://salacgriva.biblioteka.lv/Alise/v11/home.aspx dzēst
Aiviekstes pagasta bibliotēka	kristinerulle@inbox.lv	Plaviju novads	Vietējas nozīmes bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v11/home.aspx dzēst
Aizkraukles 1. vidusskolas bibliotēka	sofija.marjana@inbox.lv	Aizkraukles novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v9/home.aspx dzēst
Aizkraukles novada ģimnāzijas bibliotēka	zanesirmace@gmail.com	Aizkraukles novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v17/home.aspx dzēst
Aizkraukles pagasta bibliotēka	aizkraukles.pag.bibl@inbox.lv	Aizkraukles novads	Vietējas nozīmes bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v17/home.aspx dzēst
Aizkraukles pilsētas bibliotēka	biblioteka@aizkraukle.lv	Aizkraukles novads	Reģionu galvenās bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v17/home.aspx dzēst
Aizkraukles pilsētas sākumskolas bibliotēka	anjalepina@inbox.lv	Aizkraukles novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v18/home.aspx dzēst
Aizputes pagasta bibliotēka	olga.maler@inbox.lv	Aizputes novads	Vietējas nozīmes bibliotēkas	http://leppaja.biblioteka.lv/Alise/v19/home.aspx dzēst
Aizputes pilsētas bibliotēka	aiz-bibl@inbox.lv	Aizputes novads	Vietējas nozīmes bibliotēkas	http://leppaja.biblioteka.lv/Alise/v4/home.aspx dzēst
Aizputes pamatskolas bibliotēka	aizputes.psk.bibl@elgavasnovads.lv	Jelgavas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://ilgava.biblioteka.lv/Alise/v22/home.aspx dzēst
Alfrēda Kalniņa Cēsu mūzikas vidusskolas bibliotēka	rasma2006@inbox.lv	Cēsu novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://cesis.biblioteka.lv/Alise/v48/home.aspx dzēst
Alojas Ausekļa Vidusskolas bibliotēka	aiwanem@inbox.lv	Alojas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://limbazi.biblioteka.lv/Alise/v33/home.aspx dzēst
Alojas pagasta Ungurpilis bibliotēka	leva.praulina@inbox.lv	Alojas novads	Vietējas nozīmes bibliotēkas	http://limbazi.biblioteka.lv/Alise/v10/home.aspx dzēst
Alojas pilsētas bibliotēka	biblioteka@aloja.lv	Alojas novads	Vietējas nozīmes bibliotēkas	http://limbazi.biblioteka.lv/Alise/v20/home.aspx dzēst
Aisungas novada bibliotēka	cinula@apollo.lv	Aisungas novads	Vietējas nozīmes bibliotēkas	http://kuldiga.biblioteka.lv/Alise/v24/home.aspx dzēst
Aisungas vidusskolas bibliotēka	vinetaling@inbox.lv	Aisungas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://kuldiga.biblioteka.lv/Alise/v24/home.aspx dzēst
Aisvīķu bibliotēka	aisvibib@biblioteka@inbox.lv	Apes novads	Vietējas nozīmes bibliotēkas	http://aiukne.biblioteka.lv/Alise/v10/home.aspx dzēst
Aisvīķu pagasta Strautpils bibliotēka	strautpils.bibl@inbox.lv	Apes novads	Vietējas nozīmes bibliotēkas	http://aiukne.biblioteka.lv/Alise/v10/home.aspx dzēst
Aiķenes pilsētas bibliotēka	biblioteka@aiukne.lv	Aiķenes novads	Reģionu galvenās bibliotēkas	http://aiukne.biblioteka.lv/Alise/v10/home.aspx dzēst
Aiķenes sākumskolas bibliotēka	mar-vimba@inbox.lv	Aiķenes novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://aiukne.biblioteka.lv/Alise/v24/home.aspx dzēst
Amatas novada Drabešu internātpamatskolas bibliotēka	drab.bibl@inbox.lv	Amatas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://cesis.biblioteka.lv/Alise/v53/home.aspx dzēst
Amatas novada Ķīšu bibliotēka	anita_dajla@inbox.lv	Amatas novads	Vietējas nozīmes bibliotēkas	http://cesis.biblioteka.lv/Alise/v27/home.aspx dzēst
Amatas novada Līvū bibliotēka	livi.lidja@inbox.lv	Amatas novads	Vietējas nozīmes bibliotēkas	http://cesis.biblioteka.lv/Alise/v28/home.aspx dzēst
Amatas novada Sērmūķu pamatskolas bibliotēka	sermuksi7@inbox.lv	Amatas novads	Vispārējās un profesionālās izglītības iestāžu bibliotēkas	http://cesis.biblioteka.lv/Alise/v32/home.aspx dzēst
Andreja Upīša Strīveru bibliotēka	karmena.ulpe@inbox.lv	Aizkraukles novads	Vietējas nozīmes bibliotēkas	http://aizkraukle.biblioteka.lv/Alise/v4/home.aspx dzēst
Annas bibliotēka	bibl.anna@aiukne.lv	Apes novads	Vietējas nozīmes bibliotēkas	http://aiukne.biblioteka.lv/Alise/v5/home.aspx dzēst

2.19. att. Klientu administrācijas sadaļa

N2 CMS administrācijas modulis ir izstrādāts, izmantojot ASP.NET WebForms tehnoloģiju. Uzdevuma veikšanas ietvaros nācās integrēt „BIS ALISE” sistēmas portālu ar pārstrādāto administrācijas moduli, paredzot ASP.NET MVC un ASP.NET WebForms tehnoloģiju savstarpējo savietojamību. Izstrādātas klientu sadaļas formas pirmteksta fragments, izmantojot ASP.NET WebForms līdzekļus, ir attēlots 2.20. attēlā.

```

01 [ToolBarPlugin("CUSTOMERS", "customers",
02     "{ManagementUrl}/Customers/Default.aspx",
03     ToolbarArea.Navigation, Targets.Preview,
04     "{ManagementUrl}/Resources/icons/book_open.png", 130,
05     ToolTip = "administer customers",
06     GlobalResourceClassName = "ToolBar")]
07 public partial class Default : EditPage {
08
09     protected void Page_Load(object sender, EventArgs e) {
10         if (!IsPostBack) {
11             BindAuthorizeStatuses();
12
13             if (Request.QueryString.Count > 0) {
14                 int index;
15                 if (int.TryParse(Request.QueryString[0], out index))
16                     dgrCustomers.CurrentPageIndex = index;
17
18                 int status;
19                 if (int.TryParse(Request.QueryString[1], out status))
20                     ddlAuthorizeStatus.SelectedValue =
21                         status.ToString(CultureInfo.InvariantCulture);
22
23                 var name = Request.QueryString[2];
24                 if (!string.IsNullOrEmpty(name))
25                     txtName.Text = name;
26             }
27         }
28         BindCustomers();
29     }
30     ...
31 }

```

2.20. att. Klientu formas pirmteksta fragments

2.4. Sistēmas ieviešana un uzturēšana

Sistēmas ieviešanas procesā ir veikta kolēģu apmācība darbam ar N2 CMS administrācijas moduli. Kolēģi ir sastādījuši tīmekļa lapu saturu, ir augšupielādējuši nepieciešamos dokumentus, attēlus un izvietojuši nepieciešamo materiālu portālā. Kolēģi ir arī piedalījušies sistēmas izstrādes procesā, testējot izstrādāto sistēmas funkcionalitāti un

reģistrējot defektus „JIRA” sistēmā. Kolēģi lieto administrācijas moduli, klientu reģistrēšanai, ziņojumu nosūtīšanai izvēlētai grupai, parolu nomainai, statistikas iegūšanai par pēdējiem klientu apmeklēšanas datumiem, bibliotēku saraksta izgūšanai, kuras vēl nav joprojām ienākušas sistēmā u.c. funkcijām.

Sistēmas uzlikšana produkcijas vidē norisinājās uz atsevišķi izdalīta virtuāla servera. Uzturēšanas procesā ir izstrādāta speciāla ASP.NET bezsaistes tīmekļa lapa, kura tiek izvadīta lietotājam ienākot portālā, kurā pašreiz notiek atjaunošanas darbi, piemēram, tiek uzlikti pēdējo defektu labojumi. Dota „app_offline.htm” lapa ir parādīta 2.21. attēlā.



2.21. att. Portāla bezsaistes tīmekļa lapa

Sistēmas ieviešanas gaitā tika paredzēta automatizēto datu bāzes rezerves kopiju veidošanas funkcija. Tas bija jānodrošina atsevišķi, jo izvēlēta bezmaksas SQL Server datu bāzes versija, SQL Server Express, nenodrošina šādu funkcionalitāti. Dotam nolūkam ir izstrādāts SQL skripts, kas veic datu bāzes kopiju izveidošanu un izvietojumu noteiktajā direktorijā, kā arī komandas fails (batch), kurš pilda dota SQL skripta izsaukšanu un izveidoto datu bāzes kopiju arhivēšanu, izdzēšot iepriekšējās versijas, lai to kopijas skaits nepārsniegtu trīs kopijas. Komandas faila saturs ir attēlots 2.22. attēlā.

```

01 cd c:\backups\Database
02
03 del /Q backup3.7z
04 ren backup2.7z backup3.7z
05 ren backup1.7z backup2.7z
06
07 SETLOCAL
08
09 sqlcmd -S (local)\SQLEXPRESS -i
10 "c:\backups\Database\BisAliseBackup.sql"
11 "C:\Program Files\7-Zip\7z.exe" a -t7z backup1.7z
12 c:\backups\Database\*.bak
13 Del c:\backups\Database\*.bak

```

2.22. att. Datu bāzes kopiju veidošanas skripts

Tika veikta arī portāla veikspējas optimizācija, paredzot daļējo skatu (partial view) satura kešēšanu, kurš ir praktiski nemainīgs. Kešēšana tiek nodrošināta ar atribūta „Child-ActionCache” palīdzību. Daļējo skatu kešēšanas piemērs ir attēlots 2.23. attēlā.

```

01 [ChildActionCache(Duration = 43200)]
02 public PartialViewResult LeftMenu() {
03     var branchRoot = Find.RootItem;
04     var menuItems = branchRoot
05         .GetChildren(new MenuItemFilter(MenuType.LeftMenu))
06         .TryAppendCreatorNode(Engine, branchRoot);
07
08     var model = new LeftMenuModel(menuItems);
09     return PartialView(model);
10 }

```

2.23. att. Skatu kešēšanas piemērs kontrolieri

Papildus ir paredzēta arī bieži izgūstāmo un nemaināmo datu kešēšana. Tas tika panākts, izstrādājot sesijas un lietojuma ķeša pārvaldības klases, kurās objekts tiek ievietots kešatmiņā, ja tas netika izdarīts iepriekš. Šis princips ir parādīts 2.24. attēlā.

```

01 public class CacheManager {
02     public static T Get<T>(string cacheKey,
03         HttpContextBase context, Func<T> getItemCallback) where T : class {
04         var item = Get<T>(cacheKey, context);
05         if (item == null) {
06             item = getItemCallback();
07             if (item != null)
08                 context.Cache.Insert(cacheKey, item, null,
09                     DateTime.Now.AddDays(1), TimeSpan.Zero);
09         }
10         return item;
11     }
12 }

```

2.24. att. Keša pārvaldības klases fragments

SECINĀJUMI

Prakses ietvaros ir izstrādāts „BIS ALISE” informatīvais klientu portāls. Portāls ir uzlikts produkcijas vidē un atrodas zem adreses <http://bis.alise.lv>.

Uzskatu, ka darbs kopsummā ir veikts veiksmīgi, izpildot visus definētos uzdevumus. Ir veikta klientu reģistrēšana sistēmā, ir izstrādāts sistēmas administratīvais modulis lietotāju datu administrēšanai, ir mēģināts projektēt sistēmas arhitektūru pēc labākām praksēm, izmantojot projektēšanas šablonus, ir veikta sistēmas veiktspējas optimizācija un paredzētas iespējas gan lietotājiem, gan kolēģiem sazināties sava starpā, izmantojot iespēju sūtīt e-pasta ziņojumus, ir izstrādāta adaptīva lietojuma struktūra, ļaujot lietotājiem (administratoriem) pašiem labot, dzēst, veidot un modificēt Web lietojuma tīmekļa lapas un to saturus, kā arī ir veikta lietotāju apmācība un nepieciešamo sistēmas ieviešanas un uzturēšanas darbu izpilde. Kopsummā izdevās iegūt dinamisko un adaptīvo Web lietojumu, kuru var uzturēt paši kolēģi bez programmēšanas iemaņām un zināšanām. Portāls pilnīgi tiek pielāgots CMS administrācijas moduļa ietvaros.

Par būtisko trūkumu uzskatu portāla dizainu, kuru gribētos pārstrādāt. Diemžēl sistēmas izstrādei bija atvēlēts ļoti maz laika un panākt vienoto viedokli par sistēmas iespējamo izskatu neizdevās. Vairāku diskusiju gaitā tika nolemts atstāt tādu, kurš ir iepaticies visiem vairāk. Praktiski uzskatu, ka šādas lietas ir jārisina dizainerim un es vēlētos būt atbrīvots no šādu pienākumu veikšanas, koncentrējoties sistēmas izstrādei.

Strādājot „Tieto Latvia” esmu iepazinies ar uzņēmuma specifiku, organizāciju un darbības virzieniem. Uzskatu, ka es esmu izpildījis man uzticētos pienākumus un veiksmīgi īstenojis ražošanas praksi.

LITERATŪRA

- [1] Tieto – no datu apstrādes uzņēmuma līdz integrētu biznesa risinājumu partnerim / Internets. – <http://www.tieto.lv/par-mums/vesture/tieto-vesture>.
- [2] Tieto – Enator AB – Tieto Zviedrijas atzara izaugsmes vēsture / Internets. - <http://www.tieto.lv/par-mums/vesture/enator-vesture>.
- [3] Tieto – Vēsture / Internets. - <http://www.tieto.lv/par-mums/vesture>.
- [4] BIS ALISE – Vēsture / Internets. - <http://bis.alise.lv/Content/Index?page=23>.
- [5] BloombergBusinessweek: Company Overview of TietoEnator Financial Solutions AS / Internets. - <http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=26745087>.
- [6] Informācija jaunajiem darbiniekiem. Prezentācija. Rīga: Tieto Latvia, 2011. – 44. lpp.
- [7] Tieto – Our services / Internets. - <http://www.tieto.com/about-us/tieto-in-2-minutes/our-services>.
- [8] MSDN: Model-View-Controlller / Internets. - <http://msdn.microsoft.com/en-us/library/ff649643.aspx>.
- [9] MSDN: Using a Three-Tier Architecture Model / Internets. - [http://msdn.microsoft.com/en-us/library/windows/desktop/ms685068\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms685068(v=vs.85).aspx).
- [10] MSDN: The repository Pattern / Internets. - <http://msdn.microsoft.com/en-us/library/ff649690.aspx>.
- [11] Dapper – a simple object mapper for .NET / Internets. - <http://code.google.com/p/dapper-dot-net/>.

PIELIKUMI

LIETOTI SAĪSINĀJUMI

Saīsinājums	Angliski	Latviski
BIS	Library Information System	Bibliotēku Informācijas Sistēma
CRM	Customer Relationship Management	Klientu saistību pārvaldība
EBS	E-Business Suite	E-Biznesa komplekts
SQL	Structured Query Language	Strukturēta vaicājumvaloda
ERP	Enterprise Resource Planning	Uzņēmuma resursu plānošana
MVC	Model-View-Controller	Modelis-skats-kontrolieris
ORM	Object-relational mapping	Objektu-relāciju kartēšana
OOP	Object-oriented programming	Objektu-orientēta programmēšana
JPA	Java Persistence API	Java glabātuves saskarne