KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
VERSLO INFORMATIKOS KATEDRA

Romualdas Stonkus

GEDIMŲ ŠALINIMO SISTEMOS PROJEKTAVIMAS IR JOS PATIKIMUMO NUSTATYMAS

Magistro darbas

Recenzentas
prof. dr. R. Butleris

2007-05-25

Vadovas
doc. V. Pilkauskas
2007-05-25

Atliko
IFN 5/1 gr. stud.
R. Stonkus
2007-05-25

KAUNAS, 2007

SUMMARY

Fault management system reliability research

Today telecommunication operators provide various voice, networking and media services for business and public services. Some of these services are business and even life critical, therefore timely support of the telecommunication services becomes a very important part of the operator business. Systems, which are used to track and manage service related incidents often do depend on the same services as well, therefore those systems must be designed and developed in a way that they can work and provide necessary information even in case of critical situations, so that the support personnel can work with incident information and solve the issues in the timely fashion.

The incident management system is understood as a system that integrates together many systems like customer relationship management (CRM), network information system (NIS), payment system and field work management related system. It is obvious that the infrastructure architecture has great impact in the availability solution, but if the application design is based only on the fact that underlying architecture is good enough (it might be clustered, duplicated, backed up, etc.), then it is a great risk that application might fail under critical situation.

The incident management system will more likely to survive critical situation, if:

- Prime design principles are based on service oriented architecture (SOA) principles
- Integration between the systems is based on the asynchronous messaging pattern
- Possible latency and delay issues are included into availability evaluations
- Each software node is designed as autonomous as possible
- Client node software does not relies on the central storage and is able to exchange information with other client nodes
- Availability network is defined and network node measurement is in place
- Business procedures for handling critical situations are present and executable

TURINYS

Lentelių sąrašas

Paveikslėlių sąrašas

ĮŽANGA

Šiandien telekomunikacijų operatoriai teikia įvarias paslaugas: balso, interneto, daugialypės terpės. Šiomis paslaugomis verslo įmonės ir valstybės institucijos naudojasi teikdamos savo verslo paslaugas. Ryšium su tuo, operatorių teikiamos paslaugos tampa kritinės verslui, o tam tikrais atvejais – ir gyvybei. Dėl šios priežasties paslaugų kokybė ir efektyvus gedimų šalinimas tampa svarbi telekomunikacijos operatorių veiklos dalimi. Siekiant užtikrinti efektyvų gedimų šalinimo procesą yra diegiamos įvairios sistemos tinklo elementų apskaitai, tinklo monitoringui, darbo laiko valdymui, klientų informacijos valdymui ir panašiai. Visos šios sistemos dažnai apjungiamos į vieną – incidentų valdymo sistemą.

Tuo pat metu, visos minėtos sistemos, kurios užtikrina gedimų identifikavimą bei gedimų šalinimo procesų valdymą pačios priklauso nuo telekomunikacinių paslaugų bei infrastruktūros architektūros. Taigi, tinklo gedimas gali įtakoti ir incidentų valdymo sistemos veiklą ir blogiausiu atveju padaryti ją neveiksnia tuo metu, kai ji tampa labiausiai reikalinga.

Kadangi incidentų valdymo sistemos iš esmės yra sistemos integruojančios kitas, jau egzistuojančias sistemas, su tam tikru papildomu funkcionalumu – tai integruojamųjų sistemų pasiekiamumas, veikimo faktoriai taip pat įtakoja ir incidentų valdymo sistemas. Patys incidentai gali būti labai įvairaus pobūdžio – pradedant vienetiniais nusiskundimais ir baigiant masiniais gedimais, kurie įtakoja daug ir įvairių klientų. Savo ruožtu kiekvienai paslaugai ir klientui taikomos skirtingos priežiūros sutartys (SLA) bei incidentų sprendimo procedūros. Incidentų kiekis, bei dažnumas kelia papildomus reikalavimus sistemų integracijai, nes būtina užtikrinti, kad smarkiai padidėjus apkrovai ir taip sudėtinga situacija nepablogės dar labiau.

Taip pat svarbu sutarti dėl tokios sistemos pasiekiamumo vertinimo laike kriterijų, nes gali būti, jog neveikiant kažkuriai sistemos daliai, sistema iš esmės lieka funkcionali ir praranda tik dalį papildomų savybių, kurios nėra kritinės ir galima laikyti jog sistema veikia ir užduotis su ja galima atlikti.

Šio tiriamojo darbo objektas – TEO LT, AB incidentų valdymas. Ši kompanija pasirinkta dėl keleto priežasčių:

- Incidentų sprendimui naudojamos kelios sistemos;
- Sistemų platforma ir architektūros, bei galimybės yra skirtingos;
- Kiti operatoriai naudoja kitokias sistemas, bei incidentų valdymo procesai yra skirtingi, todėl kurti universalią sistemą gali būti neracionalu;

Darbo tikslai:

- Ištirti pasiekiamumo vertinimo kriterijus, bei galimus jų matavimo principus

- Palyginti galimus sistemos architektūros variantus ir jų atitikimą sprendžiant pasiekiamumo bei našumo problemas
- Ištirti tolimesnes sistemos plėtros galimybes pajungiant papildomas sistemas, kurios reikalingos incidentams spręsti, bei galimus kitus būdus sistemai tobulinti
- Ištirti sistemos panaudojamumą esant įvairiems kritiniams scenarijams

Šio darbo metu kuriama incidentų valdymo sistema, kuri apjungtų egzistuojančias klientų valdymo (CRM), tinklo informacijos (NIS), tinklo valdymo (NMS), darbo resursų valdymo (WFM), apskaitos bei kitas kompanijoje naudojamas sistemas.

1. APŽVALGINĖ DALIS

1.1. Incidentų valdymas

Pasaulyje yra įvarių sistemų skirtų gedimams šalinti (angl. *„Fault management system“* (1)), registruoti gedimus (angl. *„Service desk system“* (2)) ir panašiai. Pirmųjų užduotis registruoti gedimus, jei tokie įvyksta ir pranešti apie juos ir, jei įmanoma, užregistruoti informaciją apie problemą. Antrųjų – registruoti faktus apie gedimus, sekti jų vykdymą, bei atlikti vėlesnę analizę skirtą kurti prevencinius veiksmus. Tačiau incidento sprendimas dažniausiai yra daugelio veiksmų seka. Tokie veiksmai gali būti:

- Kliento informacijos patikrinimas
- Patikrinimas ar klientui neišjungta paslauga dėl skolų
- Incidento registravimas
- Kliento paslaugų, bei tinklo informacijos surinkimas
- Įrangos patikrinimas realiame laike
- Darbo paskyros registravimas
- Problemos sprendimo patikrinimas
- Incidento uždarymas

Nėra sistemos, kur visos šios funkcijos būtų realizuotos kaip viena programa, nes iš esmės kalbama apie labai skirtingą funkcionalumą, skirtingus tikslus keliamus atliekant vieną ar kitą veiksmą, bei skirtingus verslo procesus skirtingose įmonėse. Sukurti vieną programą visiems operatoriams būtų sudėtinga, nes tinklo elementų konfigūravimas atliekamas naudojant specifinę gamintojo teikiamą programinę įrangą. Tipinis funkcijų pasiskirstymas sistemose parodytas 1 lentelėje.

Lentelė 1. Funkcijų pasiskirstymas sistemose

Funkcija	Sistema tipiškai atliekanti funkciją
Kliento informacijos patikrinimas	CRM
Skolų tikrinimas	Finansų
Incidento registravimas, uždarymas, sekimas	HD
Įrangos patikrinimas	Priklauso nuo konkrečios įrangos ir tiekėjo
Darbo paskyros registravimas	WFM

1.2. Sistemos patikimumas

1.2.1. Patikimumo apibrėžimas ir skaičiavimas

Incidentų valdymo sistemoms yra labai svarbūs pasiekiamumo ir patikimumo faktoriai.

Pasiekiamumas pagal ITIL apibrėžiamas kaip „IT paslaugos ar komponento gebėjimas atlikti savo funkciją tam tikru metu arba per tam tikrą laiko tarpą“ (3).

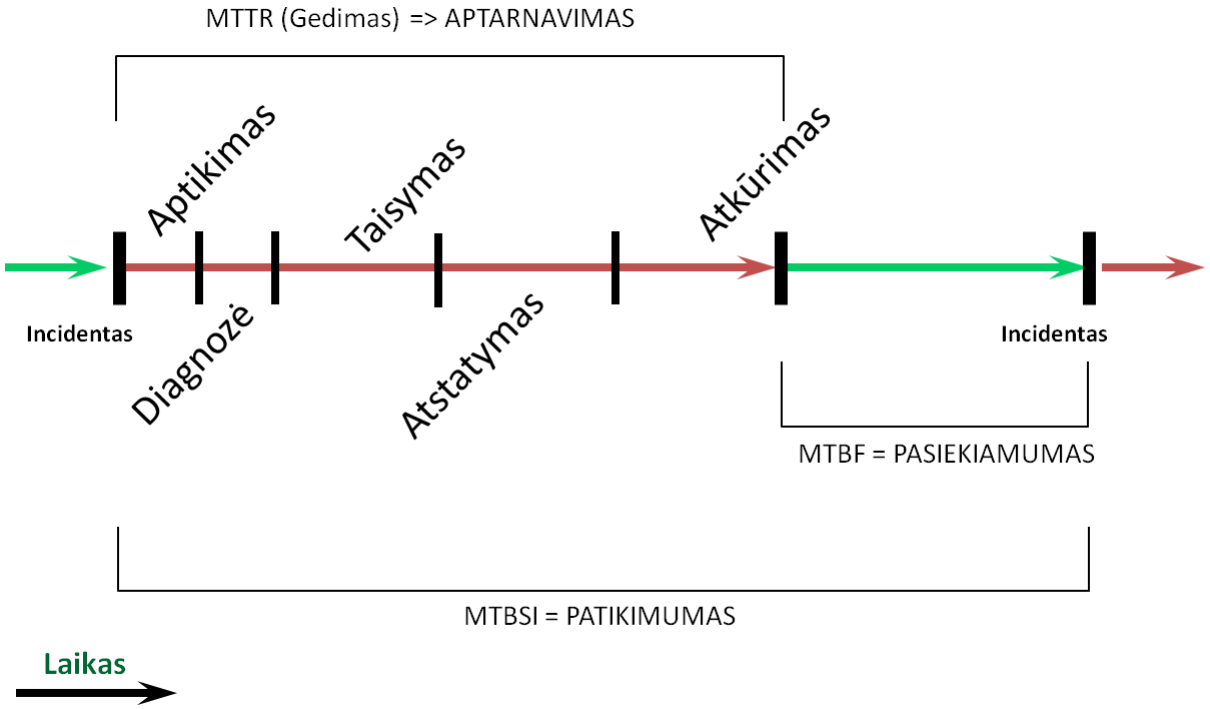
Pasiekiamumas priklauso nuo:

- Kiekvieno sistemos komponento pasiekiamumo;
- Atsparumo gedimams;
- Priežiūros ir palaikymo kokybės;
- Duomenų pasiekiamumo, saugumo ir integralumo;

Sistemos patikimumas ITIL apibrėžiamas kaip: „nepriklausomybė nuo veiklos sutrikimų“ (3) – ir savo ruožtu yra nusakoma:

- Kiekvieno komponento patikimumu;
- Atsparumu gedimams, t.y. sugebėjimą užmaskuoti komponento gedimą ir toliau tęsti verslo operacijas;

Schema iliustruojanti incidento, pasiekiamumo ir patikimumo savokas pateikta paveikslėlyje žemiau (Paveikslas 1):



Paveikslas 1. Patikimumo matavimas.

Kadangi incidentų valdymo sistema iš esmės yra sistema, kuri kartu su savo paslaugomis pateikia agreguotas aukščiau minėtų sistemų paslaugas – tai tokios sistemos veikimas, pasiekiamumas, incidento apdorojimo trukmė pradeda priklausyti nuo to, kaip veikia kiekviena iš minėtų sistemų ir kokiu būdu jos tarpusavyje yra apjungtos. Tai reiškia, jog parinkus netinkamus sistemų integravimo principus, sistema bus praktiškai neįmanoma pasinaudoti.

Kaip pavyzdį galima panagrinėti situaciją, kurio metu įvertinsime sistemos pasiekiamumą vienu iš blogesnių atvejų. Tegul vienos sistemos pasiekiamumas yra skaičiuojamas pagal formulę 1.1 (4).

$$A = \frac{MTBF}{MTBF + MTBR} \qquad \textbf{1.1}$$

Jeigu sistemos yra sujungtos nuosekliai ir įvykus nenumatytam atvejui užduoties vykdymas yra nutraukiamas, o kiekvienos sistemos pasiekiamumas yra lygus 0.95 – tai skaičiuojant bendrą pasiekiamumą pagal formulę 1.2:

$$A = \prod_{n=1}^{5} A_n \qquad \textbf{1.2}$$

gautumėme, kad bendras sistemos pasiekiamumas yra ~0.77 arba kitaip tariant vidutinė neveikimo trukmė (vertinant, kad sistema turėtų veikti 24 valandas per parą, visą savaitę) būtų ~5.5 valandos per parą. Netgi tuo atveju, jei kiekvienos iš sistemų pasiekiamumas yra 0.995, bendras tokios sistemos

pasiekiamumas yra ~0.975, o tai yra 35 minutės parai arba ~4 valandos per savaitę, arba ~2 dienos per mėnesį.

Matyti, jog jei norime, kad sistema leistų vykdyti užduotis net ir kritinių situacijų metu – reikia galvoti apie tinkamos incidentų valdymo, bei integravimo architektūros parinkimą, t.y. kurti sistemą taip, kad ji būtų kaip įmanoma atspari išorinių gedimų įtakai.

1.2.2. Pasiekiamumo stebėjimas

Netgi, jei pavyktų sukurti infrastruktūrą, bei sistemą, kuri veikia stabiliai net esant pačiom kritiškiausiom sąlygom – būtina numatyti procedūras ir priemones kaip bus identifikuojamos neveikiančios dalys ir kas tokiu atveju bus daroma.

Kaip paprastą tokio atvejo pavyzdį galima panagrinėti situaciją, kai sugenda serverio kietasis diskas. Jeigu tai yra pavienis diskas – tai OS praneš, jog negali prieiti prie duomenų, o jeigu tai sisteminis diskas – tai OS nustos veikti. Pirmuoju atveju mes pamatysime pranešimus įvykių žurnale ir galbūt negalėsime prieiti prie būtinų duomenų. Antruoju atveju – mes negalėsime prieiti prie duomenų ir/arba pasinaudoti bet kokiomis paslaugomis iš minėtojo serverio.

Tuo tarpu jeigu diskas yra RAID 1+0, arba RAID 1, arba RAID 5 masyvo dalis – tai sistema toliau turėtų veikti be sutrikimų, tačiau pati savaime situacija yra grėsminga ir normaliomis sąlygomis reikėtų imtis atitinkamų priemonių. Dažnai OS tokie gedimai tiesiogiai nėra registruojami, o tai atlieka gamintojų papildoma programinė įranga. Taigi, galima teigti, jog netgi naudojant RAID masyvus visviena būtina sekti jo būseną ir esant neigiamiems pakeitimams imtis atitinkamų veiksmų, pavyzdžiui pakeisti sugedusį diską.

Analogiška situacija yra ir su bet kuria kita sistemos dalimi: duomenų bazėmis, web serveriais, aplikacijos komponentais, windows servisais, ir panašiai. Sukūrus sistemą, kuri išgyvena programinės įrangos gedimus – reikia numatyti, jog esant tokiems sutrikimams turi būti imtasi atitinkamų priemonių, todėl greta kiekvieno numatomo komponento būtina papildomai suprojektuoti šiuos elementus:

- Kaip, kokiomis priemonėmis ar būdais bus identifikuojamas komponento gedimas;
- Veiksmų planas ir procedūra, kurios reikėtų laikytis norint pašalinti minėtą gedimą;

1.2.3. Pasiekiamumo valdymo metodikos

Yra daug metodikų, kurios gali būti taikomos vertinant sistemų pasiekiamumą, bei taikomus sistemų valdymo procesus. Keletas jų išvardinta žemiau:

- Komponento gedimo įtakos analizė (CFIA)

- Gedimų meždio analizė (FTA)
- CRAMM

Kiekviena iš minėtų metodikų turi savo privalumus ir trūkumus, kurie pavaizduoti lentelėje žemiau:

Lentelė 2. Pasiekiamumo valdymo metodikos.

Metodika	Planavimas	Gerinimas	Ataskaityba
CFIA	X	X	X
FTA	X	X	
CRAMM	X	X	

Toliau darbe taikysime patį paprasčiausią CFIA modelį, kurio principas yra surašyti matuojamus elementus, bei jų įtaką paslaugoms arba funkcionalumui.

1.3. Sistemų integracija

1.3.1. Integravimo šablonai

Integruojant sistemas reikia atsižvelgti į galimus integracijos būdus bei priemones. Integruojant sistemas reikia nuspręsti koks mūsų tikslas: ar mes norime apjungti duomenis, ar aprašyti verslo procesus ar tiesiog atvaizduoti viską viename ekrane. Vėliau galima nuspręsti kokiais būdais šias sistemas apjungti, bei kokias topologijas galima būtų taikyti idant pasiekti norimus rezultatus.

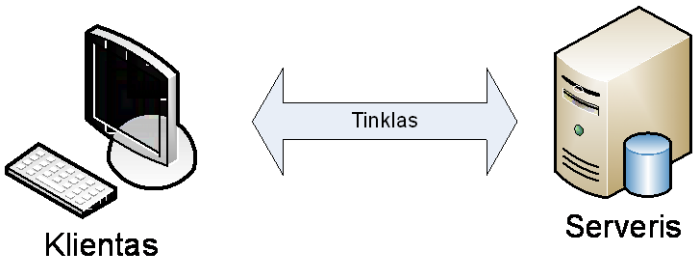
Tipiniai integravimo problemų sprendimo scenarijai aprašomi integravimo šablonais, kuriuos galima suskirstyti į šias grupes:

- Integravimo lygmenų šablonai: esybių, procesų ir portalų,
- Sistemų apjungimo šablonai: duomenų, funkcinis, vartotojo sąsajos ir orientuotas į paslaugas,
- Integravimo topologijos: „taškas-į-tašką“, pranešimų maršrutizatorius, pranešimų kanalas ir publikavimas-prenumerata,
- Papildomi integravimo šablonai: filtrai, šliuzai ir t.t. (5).

Kuriant incidentų valdymo sistemos integravimo architektūrą – galima atsižvelgti ir atitinkamai taikyti šiuos šablonus, tačiau reikia atsižvelgti į jų tinkamumą siekiant užtikrinti norimą pasiekiamumą.

1.3.2. Integravimas ir pasiekiamumas

Net ir analizuojant atskirą „klientas-serveris“ (Paveikslas 2) tipo sistemą matyti, jog yra mažiausiai trys elementai, kuriems nustojus veikti – nustos veikti pati sistema. Tai kliento sistema, tinklas, serverinė dalis.

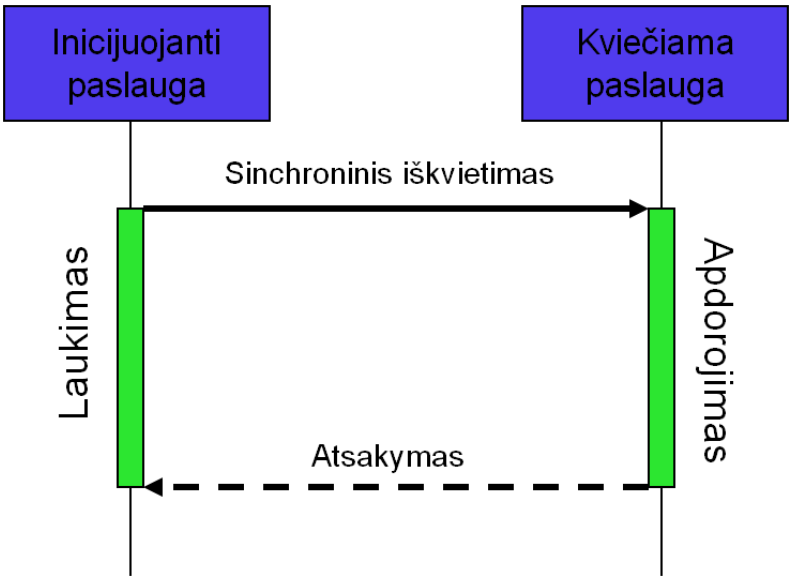


Paveikslas 2. Klientas-Serveris architektūra.

Rizikas, kad kažkuris elementas neveiks galima mažinti taikant infrastruktūros sprendimus. Turėti ne vieną, o keletą kliento kompiuterių, dubliuotą tinklo bei serverių infrastruktūrą. Tačiau net ir dubliuotos sistemos genda, tiesa su mažasne tikimybe.

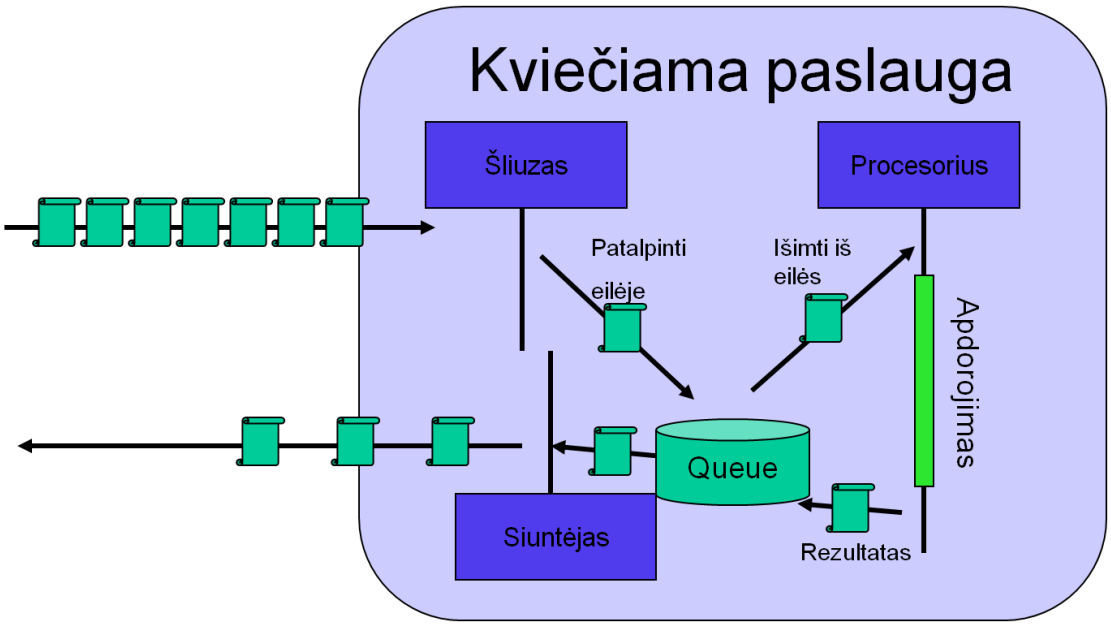
Jei analizuoti kelių loginių lygmenų sistemą – tai tokių elementų skaičius padidėja iki 5. Incidentų valdymo procese dalyvauja penkios sistemos, neskaitant pačios incidentų valdymo sistemos sudėtinių elementų. Šių sistemų išlygiagretinti negalima, nes kiekviena jų atlieka savo funkcijas ir vienos veiksmai priklauso nuo kitos sistemos grąžinamų rezultatų. Taip pat, skirtingos sistemos apdoroja skirtingą transakcijų kiekį.

Šiuo metu integracijai dažnai tiesmukai naudojami, tačiau kalbant apie incidentų valdymos sistemas – tiesioginis sistemų sujungimas web servisais gali būti nenaudingas, nes atsiranda priklausomybė tarp vykdymo laikų, t.y. jei serveris ilgai atlikinėja užduotį – atitinkamai ilgai blokuojami resursai ir klientinėje sistemoje (Paveikslas 3), o taip pat neįmanoma valdyti apkrovą serveryje.



Paveikslas 3. Sinchroninis iškvietimas.

Šią situaciją galima pakeisti naudojant bendravimą pranešimais, kai į kviečimą paslaugą siunčiami pranešimai yra rašomi į eilę, o serviso branduolys – skaito pranešimus tokiu greičiu, kokiu įmanoma esamoje sistemoje, bei patalpina rezultatus atgal į eilę (Paveikslas 4).



Paveikslas 4. Pranešimai ir jų apdorojimas.

Taikant minėtą schemą dingsta priklausomybė tarp paslaugos greičio ir užklausų kiekio, be to, esant būtinybei, kiekvieną iš elementų galima dauginti ir tuo būdu užtikrinti norimas paslaugos laikines savybes. Tačiau tuo pat metu keičiasi įprastas programavimo modelis, bei dingsta bet kokia galimybė kontroliuoti nutolusios sistemos veikimą. Dėl šios priežasties, kai sistemos bendrauja tarpusavyje

pranešimais dažnai naudojamos papildomos programinės priemonės užtikrinančios būtiną infrastruktūrą pranešimų skaičiavimui, laiko matavimui, būsenos užtikrinimui ir panašiai.

Atsižvelgiant į aukščiau išdėstytą medžiagą galima konstatuoti, kad:

- Vadovaujantis orientuoto į paslaugas apjungimo šablonu galima logiškai izoliuoti sistemas vieną nuo kitos. Tokiu būdu veikimas tampa mažiau priklausomas nuo kitos sistemos pokyčių;
- Taikant pranešimų asinchroninio apdorojimo šabloną, galima spręsti atskiros sistemos pasiekiamumo ir apkrovos paskirstymo klausimus;
- Kontroliuojant pranešimų perdavimą tarp sistemų atskiru procesu galima realizuoti būtinų veiksmų atlikimą nustatyta tvarka, bei kontrolę;
- Vertinant sistemos pasiekiamumą reikia įvertinti ir galimus delsimus, kurie būtų didesni nei leistina;
- Bendruoju atveju programavimo sudėtingumas nemažėja, o netgi priešingai – didėja.

1.4. Projekto uždaviniai

Remiantis aukščiau išdėstyta informacija nuspręsta, jog reikia sukurti incidentų valdymo sistemą. Šiam projektui keliami šie uždaviniai:

- Išanalizuoti ir aprašyti keliamus reikalavimus;
- Išsiaiškinti projekto ypatumus susijusius su kritinių situacijų bei SLA valdymu;
- Aprašyti testavimo ir pritaikyti testavimo metodologiją;
- Aprašyti incidentų sprendimo procesus;
- Aprašyti sistemos architektūrinį modelį;
- Aprašyti sistemos objektų modelį;
- Aprašyti sistemos duomenų struktūrų modelį;
- Aprašyti vartotojo sąsajas;

Toliau darbe yra aprašomas pats projektas.

2. PROGRAMINĖS ĮRANGOS PROJEKTAS

2.1. Sistemos paskirtis

Projekto paskirtis sukurti sistemą incidentams registruoti (IVS). Šios IS tikslas – registruoti incidentus, surinkti būtiną informaciją reikalingą jų sprendimui, užtikrinti jų sprendimą, bei pateikti statistinę informaciją vadovams. Pagrindinės funkcijos būtų šios:

- Incidentų registravimas;
- Informacijos apie incidentą agregavimas iš įvairių IS;
- Incidentų sprendimo proceso užtikrinimas;
- Informacijos apie einamuosius incidentus pateikimas;
- Incidentų statistinės informacijos pateikimas analizei;
- Užtikrinti galimybę registruoti, bei spręsti incidentus esant kritinėms situacijoms, kai viena arba kelios integruojamos sistemos neveikia;
- Sistemos gyvybingumo registravimas ir pranešimai apie pasikeitusią sistemos būklę;

2.2. Klientas, bei pagrindinis sistemos naudotojas

TEO LT, AB

Savanorių 28, Vilnius

2.3. Sistemos apimtis

Kuriant sistemą turėtų būti realizuotos šios dalys:

- Klientinė programa TEO LT, AB darbuotojams, incidentams registruoti;
- Tinklinė programa TEO LT, AB klientams, kurios pagalba galėtų registruoti incidentus;
- Sąsajos prieigai prie egzistuojančių IS;
- Incidentų sprendimo proceso, bei kontrolės realizacija;
- Ataskaitos apie einamuosius gedimus;
- Statistinė informacija apie anksčiau registruotus incidentus;
- Nustatytos procedūros kritiniams atvejams;
- Sukurta automatinių testų infrastruktūra;
- Sistemos patikimumo matavimo procedūros bei priemonės;
- Numatyti galimi tolimesni sistemos vystymo būdai;

Į projekto apimtis neįtraukiama:

- Integruojamųjų sistemų perrašymas, taisymas ar sukūrimas;
- Klientų savitarnos puslapių sukūrimas ar perrašymas;
- Programų diegimas;

2.4. Reikalavimai sistemai

2.4.1. Pradiniai reikalavimai

Pradžioje surinkti reikalavimai pateikti priede (1 PRIEDAS. Pradiniai reikalavimai). Šie reikalavimai išgryninti išnagrinėjus tiekėjams pateiktą pirminį užklausimą. Vienok akivaizdu, jog šie reikalavimai labai paviršutiniškai apibūdina pačią sistemą ir reikalavimus jos veikimui, todėl buvo atliktos papildomos apklausos.

Apklausų metu buvo apklausti šie žmonės:

Lentelė 3. Apklausoje dalyvavę asmenys

Vardas Pavardė	Pareigos
Almantas Klimas	IT skyriaus vadovas
Gražvydas Varkalys	Tinklo informacinės sistemos sąvininkas
Gintarė Žilinskienė	Procesų departamento kokybės skyriaus vadovė
Vytautas Bučinskas	Procesų departamento direktorius

Šie žmonės buvo parinkti, nes jie gali pateikti papildomą informaciją apie:

- gedimo šalinimo procesą;
- darbuotojų elgsenos apibūdinimą su dabartinėmis sistemomis;
- tikimąsi gedimų šalinimo proceso eigą;
- gedimų šalinimo sistemos tikslus;
- papildomus reikalavimus programinei ir aparatinei įrangai;

Iš papildomo interviu tapo aiškūs papildomi faktai, tokie kaip:

- Reikalavimai programinei įrangai;
- Vartotojų, vadovų, specialistų veiklos scenarijai;
- Reikalavimai pasiekiamumui ir procedūros esant kritinėms situacijoms;
- Aprašytas gedimų šalinimo procesas ir sistemų panaudojimas procese;

Patikslinti reikalavimai yra išdėstyti kitam skyriuje.

2.4.2. Papildyti reikalavimai

Po interviu, pirminiai reikalavimai buvo išanalizuoti ir įvardinti naujai. Diskusijos metu buvo nustatyta, jog yra keletas svarbių tikslų šios sistemos sukūrimui:

- Kiekvienas incidentas turi būti užregistruotas, jie negali būti pamesti dėl jokių priežasčių;
- Didelio prioriteto incidentai (kurių SLA lygis aukštas) turi būti sprendžiami nepaisant incidentų valdymo sistemos būsenos;
- Būtina kontroliuoti incidento sprendimo eigą;
- Statistinė informacija apie gedimus gali suteikti papildomos informacijos apie gedimo priežastį;

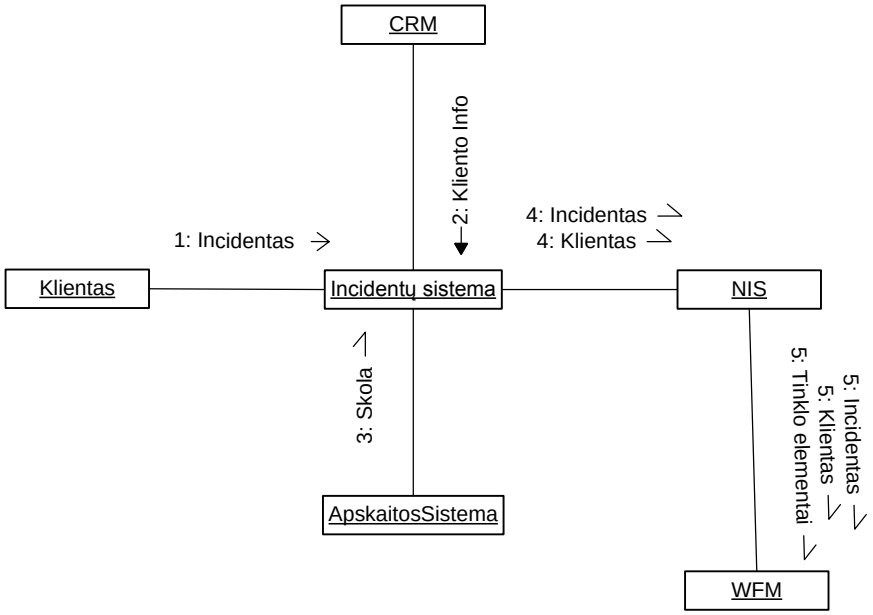
Interviu metu buvo įvardinti reikalavimai programinei įrangai. Jie yra pateikti priede (2 PRIEDAS. Reikalavimai programinei įrangai).

Detalizuoti reikalavimai išdėstyti priede (3 PRIEDAS. Papildyti reikalavimai sistemai).

Diskusijų metu buvo įvardinta, jog šiuo metu egzistuoja procedūra kaip pereinama prie popierinio sistemos varianto, kai nustoja veikti susijusios IS:

- Jeigu neveikia sistema, yra informuojamas pamainos viršininkas;
- Pamainos viršininkas patikrina savo kompiuteryje;
- Jeigu problema pasitvirtina – incidentai yra registruojami ant popieriaus arba Excel failuose;
- Pamainos viršininkas informuoja administratorių apie gedimą;
- Atstačius sistemos veiklą – popieriuje surašyti gedimai suvedami į sistema;

Tikslinant reikalavimus buvo patikslinta ir incidentų valdymo proceso schema atsižvelgiant į naudojamas sistemas (Paveikslas 5). Kaip matyti, daugumo procesų gali būti asinchroniai, t.y. užregistravus incidentą, visa likusi informacija gali būti apdorojama tada, kai tai yra įmanoma.



Paveikslas 5. Gedimų šalinimo proceso schema.

Tolimesnės analizės metu įvardinti personos (aktoriai) yra įvardinti lentelėje (Lentelė 4).

Lentelė 4. Personų sąrašas.

Persona	Aprašymas
Klientas	Naudojasi TEO LT, AB paslaugomis. Jeigu jos nustoja veikti arba veikia nekokybiškai, praneša apie incidentą.
Klerkas	Atsakingas už incidentų, pranešamų telefonu ir e-paštu registravimą incidentų sistemoje, bei incidento uždarymą.
Vadybininkas	Asmuo priskirtas konkrečiam klientui (top 500). Jeigu įvyksta incidentas tokiam klientui, vadybininkas yra informuojamas. Toliau vadybininkas atsakingas už incidento sprendimo kontrolę, kokybės užtikrinimą, kliento informavimą, bei incidento uždarymą.
Specialistas	Asmuo, atsakingas už konkretaus gedimo šalinimą nuotoliniu būdu arba vietoje pas klientą.
Specialistų vadovas	Stebi bei analizuoja gedimų srautą, paskirsto konkrečias užduotis, atsakingas už svarbių užduočių sprendimą laiku, bei sudėtinių užduočių identifikavimą.

Administratorius	Prižiūri IS, užtikrinančias incidentų valdymą. Iškilus nesklandumams sistemoje, imasi juos spręsti. Stebi įvykių žurnalus, bei naudojasi papildomomis priemonėmis nesklandumams pastebėti.
Proceso vadovas	Stebi ir vertina bendrą gedimų tendenciją, sprendimo trukmes, resursų paskirstymą ir t.t. Pagrindinė užduotis – tobulinti procesą tuo būdu didinant klientų pasitenkinimą paslaugomis.

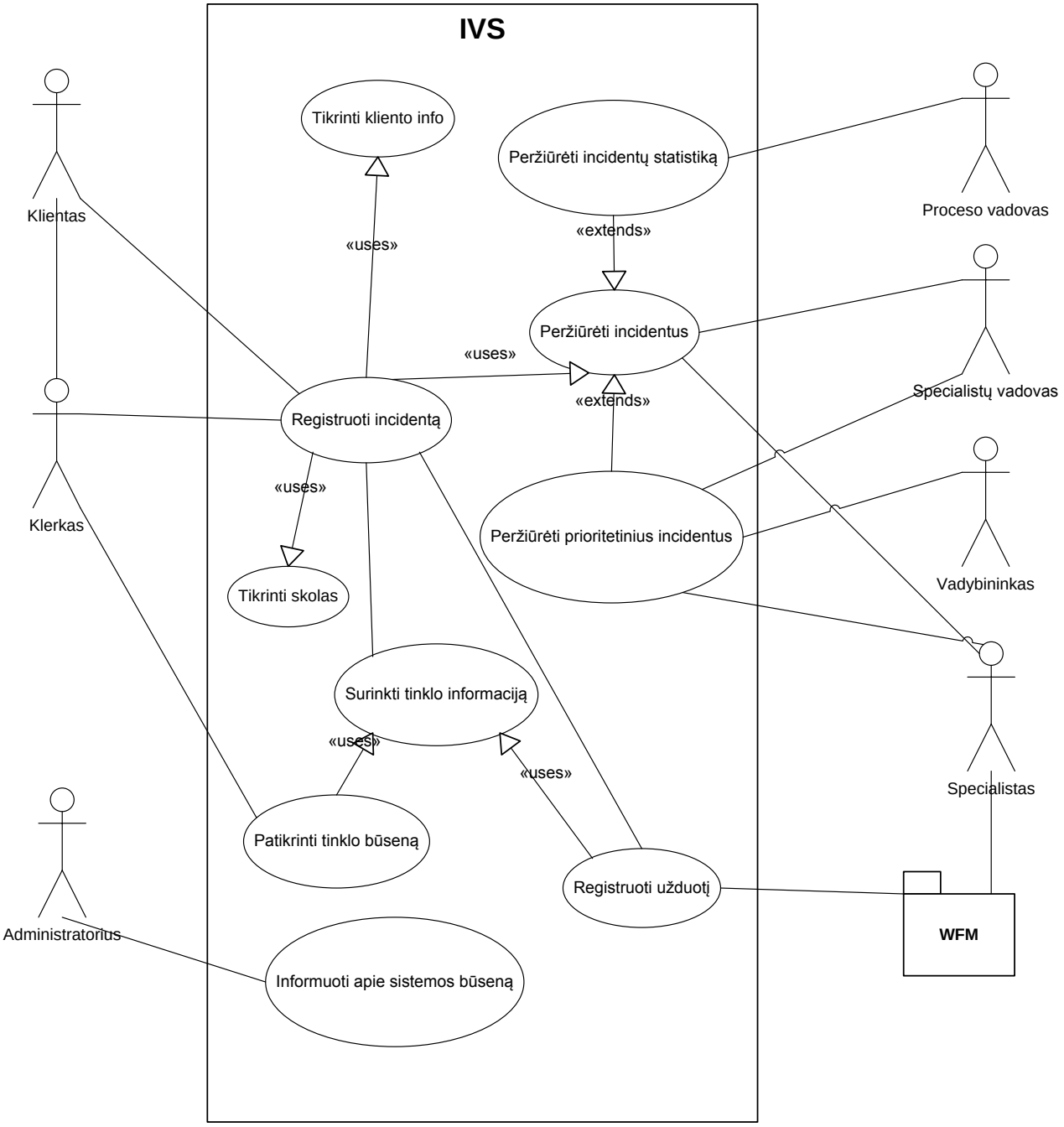
Pagal pateiktus reikalavimus identifikuoti panaudojimo atvejai pateikti lentelėje (Lentelė 5).

Lentelė 5. Panaudos atvejai.

Nr.	Panaudojimo atvejis	Aprašymas
1	Registruoti incidentą	Klientas arba klerkas registruoja incidentą
2	Tikrinti kliento informaciją	Sistema patikrina ar yra toks klientas, ar jis turi minėtą paslaugą.
3	Tikrinti skolas	Sistema patikrina ar klientas nėra skolingas už paslaugas. Jeigu skolingas – incidentas neregistruojamas.
4	Patikrinti tinklo būseną	Tam tikrom paslaugom sistema gali patikrinti liniją iki pat kliento automatiškai.
5	Surinkti tinklo informaciją	Sistema, pagal kliento pateiktą informaciją surenka informaciją apie tinklo elementus: stotį, maršrutą, portus – kurie vėliau gali būti panaudoti tikrinant paslaugą arba registruojant problemą.
6	Registruoti problemą	Sistema registruoja problemą WFM sistemoje, kuria naudojasi specialistai.
7	Peržiūrėti incidentų statistiką	Sistema parodo ilgalaikę incidentų ataskaitą.
8	Peržiūrėti incidentus	Sistema atvaizduoja incidentus, bei leidžia juos uždaryti, jei buvo išspręsti.
9	Peržiūrėti prioritetinius incidentus	Sistema parodo sąrašą einamųjų prioritetinių incidentų ir informaciją apie kiekvieną iš jų. Taip pat leidžia uždaryti incidentą, jei jis buvo išspręstas.
10	Informuoti apie sistemos būseną	Sistema parodo sistemos komponentų, bei jų tarpusavio ryšių būseną.

Visi šie panaudos atvejai yra atvaizduoti diagramoje (Paveikslas 6).

Iš diagramos matyti, jog nors specialistai iš esmės dirba su WFM sistema, kritiniais atvejais jie gali naudotis tiesiogine informacija apie incidentus iš IVS sistemos. Tas pats galioja ir kalbant apie klerkus – jiems taip pat svarbu žinoti, ar kliento incidentas jau registruotas ar ne, bei esant reikalui – eskaluoti vieną ar kitą problemą.



Paveikslas 6. Panaudos atvejų diagrama.

Iš šios diagramos ir turimų reikalavimų seka kelios išvados svarbios tolimesnei sistemos architektūrai:

- Klerko klientas turi minimaliai priklausyti nuo centrinių duomenų bazių serverių;
- Būtina turėti galimybę užregistruoti, bei išsaugoti incidento informaciją ne kliento programoje;
- Turi būti numatytas galimybė patikrinti incidentus „atsitiktiniems“ darbutojams;

Pagal šią informaciją yra detalizuoti sistemos architektūros aspektai pateikti 4-tame priede (Lentelė 15).

2.5. Projekto planas

2.5.1. Projekto užduotys

Šiam projektui buvo identifikuotos pagrindinės užduotys būtinos šiam projektui atlikti

Nr.	Užduotis	Komentarai
1	Aprašyti pagrindinius testus	Aprašydami testus įsitikinsime ar visi reikalavimai gali būti pratestuoti. Nusistatyti taikomus testavimo metodus.
2	Aprašyti sistemos architektūrą	Tikslas: tiksliau identifikuoti būsimus sistemos komponentus.
3	Aprašyti detalią sistemos architektūrą	Detalizuoti sistemos komponentų dizainą, bei patikslinti reikalavimus.
4	Aprašyti planuojamus kodo testus	Sukurti automatinių testų rinkinį rašomam kodui testuoti.
4	Atlikti programavimo darbus	Sukurti IVS programą.
5	Atlikti testus	Įsitikinti, kad sistema tenkina iškeltus reikalavimus, bei kokybinius kriterijus
6	Paruošti sistemos diegimo paketus	Sukompiliuoti sistemos diegimo paketus

2.5.2. Projekto rizikos

Projekto pradžioje įvertintos rizikos pateiktos lentelėje žemiau.

Lentelė 6. Projekto rizikos.

Nr.	Rizika	Tikimybė	Įtaka
1	Gauta sistema bus per sudėtinga administratoriams suvaldyti	Didelė	Didelė
2	P2P protokolų našumas netenkins kritiniais scenarijais	Vidutinė	Didelė
3	Pernelyg sudėtinga vartotojo sąsaja apsunkins vartotojų darbą	Vidutinė	Vidutinė
4	Vartotojai nesugebės atpažinti ir naudotis sistema kritiniais atvejais	Vidutinė	Vidutinė
5	Vadovybė reikalaus greito sprendimo	Didelė	Vidutinė
6	Projekto sudėtingumas ir kaina išaugs	Vidutinė	Vidutinė
7	Ne visi operaciniai ir kokybiniai reikalavimai	Vidutinė	Didelė

Šias projekto rizikas galima suskirstyti į dvi kategorijas:

- Technologines – susijusias su tam tikrų naujų technologijų panaudojimu;
- Projektines – susijusias su projekto valdymo aspektais: finansais, užtikrinimu, resursais ir panašiai;

Žemiau esančioje lentelėje pateiktas rizikų valdymo planas.

Lentelė 7. Rizikų mažinimo planas.

Rizika	Tikimybės mažinimas	Problemos sprendimas
1	Įtraukti į dizainą priemones sistemos komponentų gedimui identifikuoti.	Parašyti detalesnę dokumentaciją ir procedūras tipinėms probleminėms situacijoms.
2	Iš anksto atlikti testus su didele apkrova.	Mažinti perduodamą duomenų kiekį, turėti centrinę sinchronizavimui skirtą kopiją.
3	Minimizuoti vartotojo sąsajos elementų skaičių.	Pateikti pavyzdžius dažniausiai pasitaikantiems scenarijams, papildyti dokumentaciją, mokyti vartotojus.
4	Pateikti vartotojo interfeise informaciją apie	Mokyti vartotojus.

	susidariusią kritinę situaciją.	
5	Pateikti siūlomo varianto privalumus ir trūkumus, bei, svarbiausia, ilgalaikės perspektyvos privalumus.	Skaldyti projektą į mažesnius, pirma pateikiant pagrindinį funkcionalumą.
6	Projektą komponentus vertinti atskirai, kiek įmanoma detaliau. Anksčiau atlikti „proof-of-concept“ patikrinimą.	Spręsti situaciją pagal sutartą „vertės-laiko-resursų“ susitarimą. Šiuo atveju, išaugus kaštams, mažinti projekto apimtį.
7	Anksčiau aprašyti funkcinius testus, palyginti sistemos architektūrą su reikalavimais ir CFIA, peržiūrėti ar identifikuoti saugumo, priežiūros, našumo, bei pasiekiamumo reikalavimai	Vykdyti pakeitimų valdymo procedūrą

2.6. Sistemos architektūra

2.6.1. Sistemos loginė struktūra

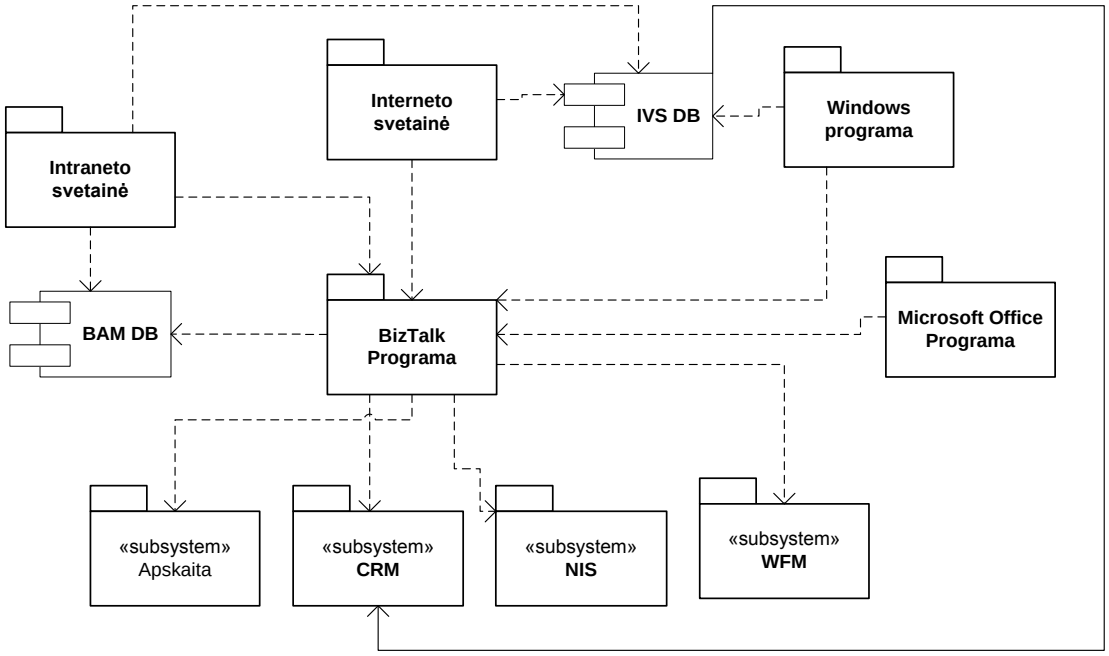
IVS – yra sudėtinė sistema, sudaryta iš keleto aplikacijų, besikeičiančių informacija apie registruotus incidentus. Taip pat IVS integruoja jau egzistuojančias sistemas: CRM, NIS, WFM, bei apskaitos.

IVS – pagrindinės sistemos dalys yra:

- Windows programa – skirta incidentų registravimui;
- Interneto svetainė – skirta klientams, incidentų registravimui ir sekimui;
- Intraneto svetainė – darbuotojams, procesų vadybininkams, etc.
- Microsoft Office formos – priemonė incidentams registruoti ne įmonės tinkle, be interneto prieigos;
- BizTalk programa – incidentų sprendimo proceso užtikrinimui;
- Išorinės sistemos;

IVS konceptualus architektūros modelis pavaizduotas paveiksle žemiau (Paveikslas 7).

Sprendžiant iš diagramos, galima nuspėti, kad tam tikra dalis verslo logikos (taisyklės, procesai, esybės) bus vienodi visoms trims klientinėms aplikacijoms. Kitais žodžiais tariant – pastarąsias galima traktuoti kaip tos skirtingas tos pačios sistemos vartotojo sąsajas.



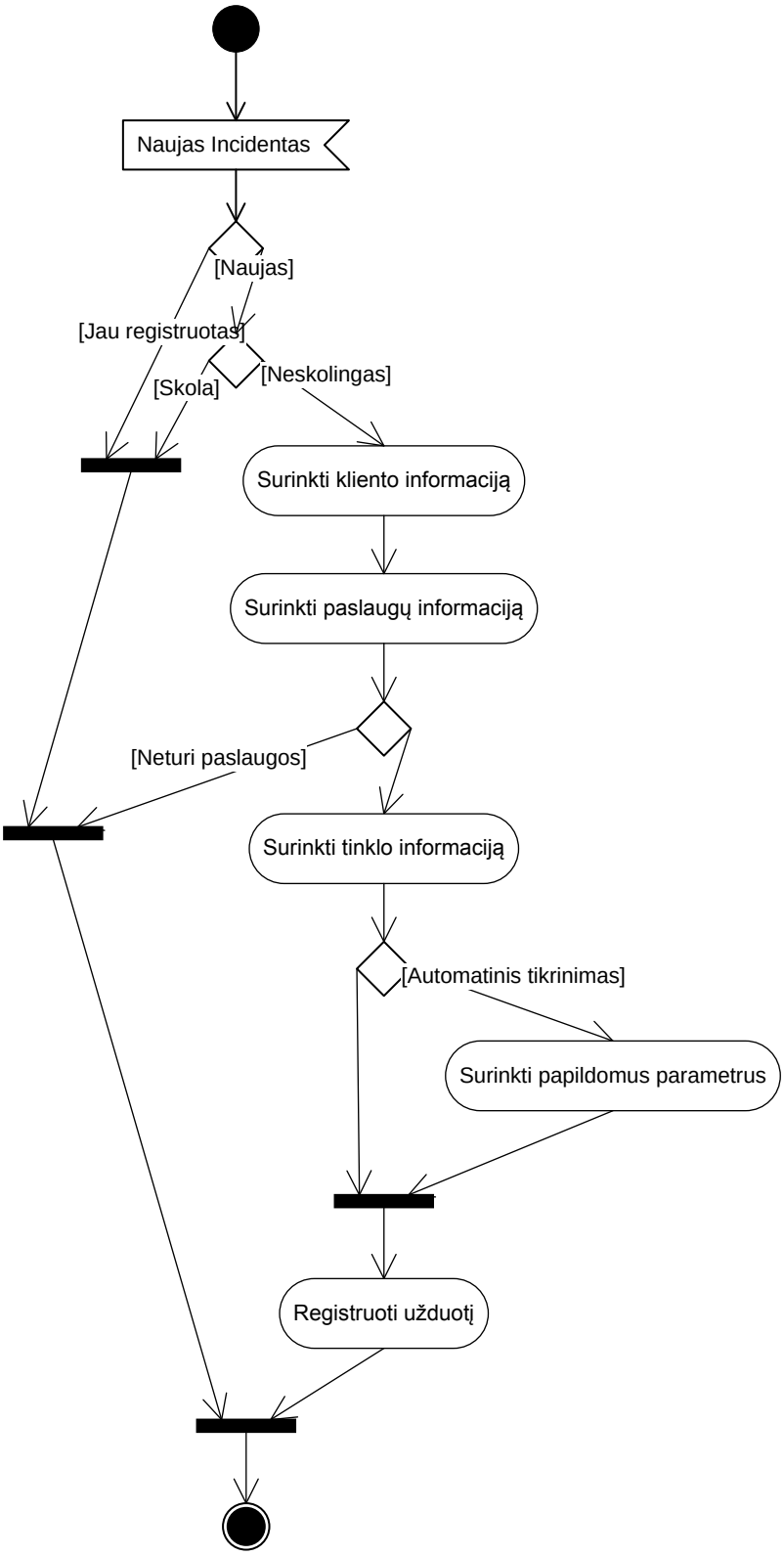
Paveikslas 7. Sistemos architektūros modelis

2.6.2. Sistemos dinaminis modelis

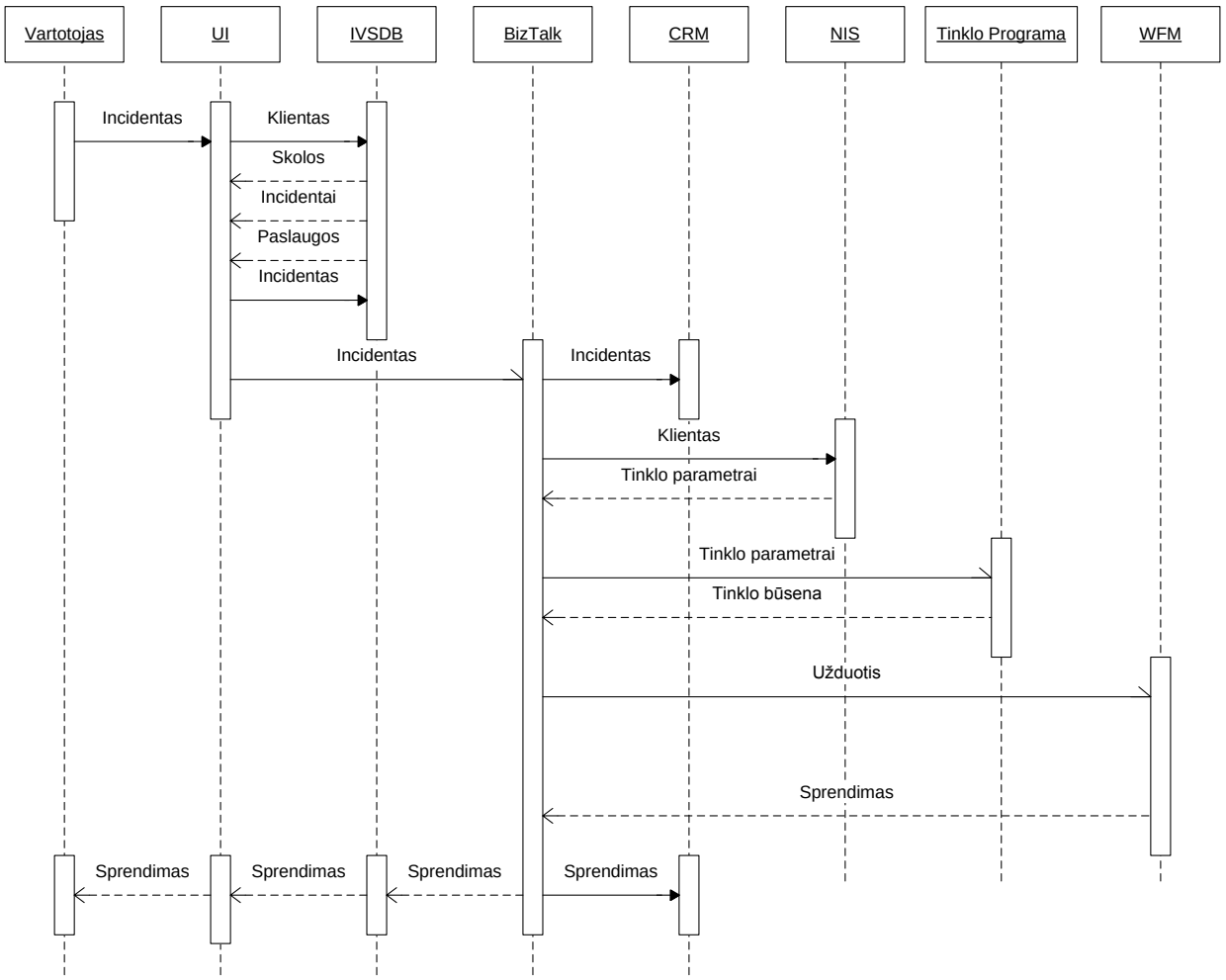
Sistemos dinaminis vaizdas aprašytas veiklos ir sekų diagramomis.

Veiklos diagramoje (Paveikslas 8) atvaizduotas tipinis incidento sprendimo procesas: nuo incidento įvykio, bei registracijos – iki galutinio sprendimo.

Sekų diagramoje (Paveikslas 9) pavaizduotas sistemų tarpusavio bendravimas, bei perduodami pranešimai. Iš diagramos matyti, jog pirminis informacijos apie klientą patikrinimas vyksta sinchroniškai (tai reikalinga klerkams, skambučių centrų atveju), todėl ši informacija turės būti kaip įmanoma arčiau vartotojo – galbūt net tame pačiame kompiuteryje. Tuo tarpu BizTalk programa – atlieka proceso koordinavimo, bei pranešimų persiuntimo kitoms sistemoms užduotis. Šis procesas vyksta asinchroniškai ir kliento veikimas nuo pastarojo neprklauso, dėl šios priežasties sumažėja vartotojo sąsajos neveikimo tikimybė, kitais žodžiais tariant, jeigu BizTalk sistema neveikia – tai neturi jokios įtakos klientinėms programoms. Tačiau tuo pat metu tai sąlygoja papildomus reikalavimus: naudoti asinchroninį pranešimų perdavimo transportą (MSMQ), arba keletą jų lygiagrečiai (Web servisai, MSMQ, failai, ir panašiai).



Paveikslas 8. Incidento sprendimo veiklos diagrama.



Paveikslas 9. Incidento apdorojimas sistemose.

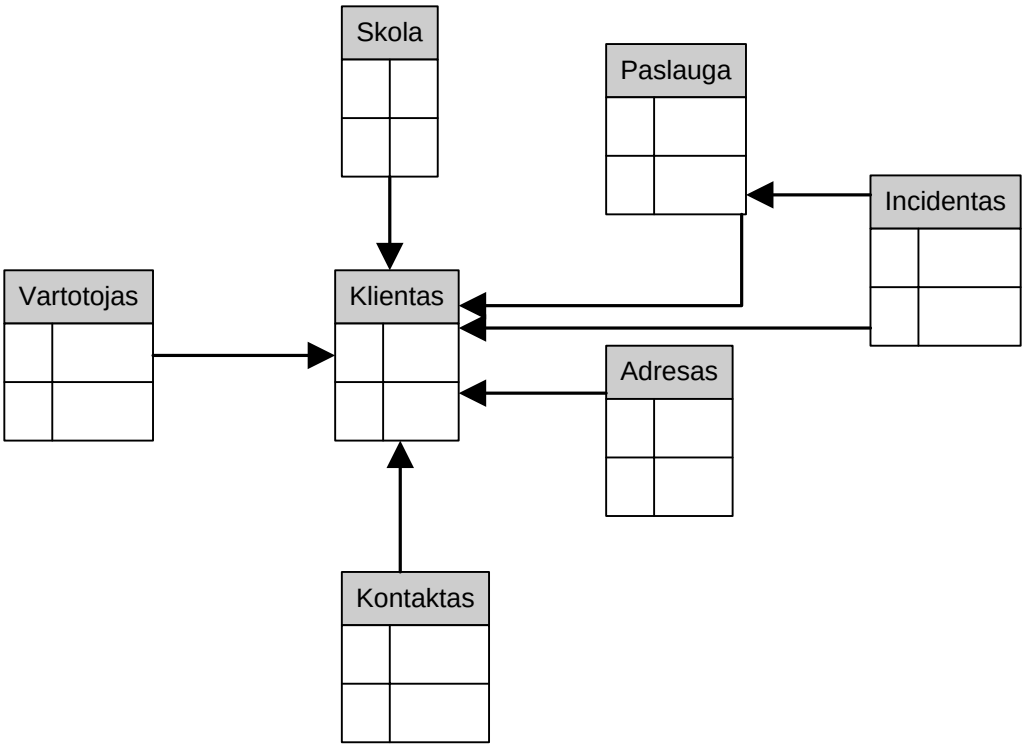
2.6.3. Konceptualus duomenų modelis

IVS sistemoje duomenys yra paskirstyti įvairiuose komponentuose, bei integruojamose sistemose. Integruojamųjų sistemų duomenų modelio šiame darbe nenagrinėsime. IVS sistemoje bus ne viena duomenų bazė, o kelios:

- Duomenų bazė windows programai;
- Centralizuota duomenų bazė;
- Interneto portalo vartotojų duomenų bazė;

Centralizuotoje ir windows programos duomenų bazėse turės būti patalpinta pati būtiniausia informacija apie klientą, paslaugas ir skolas, kad sistemos galėtų funkcionuoti sugedus toms IS, kurios yra minėtų duomenų pagrindiniai šaltiniai.

Atsižvelgiant į tai, konceptualus IVS duomenų modelis atrodo kaip pateikta paveikslėlyje, o lentelėje esančioje žemiau – aprašytas kiekviena esybė pavaizduota diagramoje.



Paveikslas 10. Konceptualus duomenų modelis.

Esybė	Aprašymas
Klientas	IVS atveju tai subjektas, kurio vardu registruojamas incidentas
Vartotojas	Prisijungimui prie savitarnos svetainės reikaling informacija
Kontaktas	Kontaktinis asmuo, su kuriuo bendraujama dėl incidento
Skola	Informacija apie kliento įsiskolinimus, jei tokie yra
Paslauga	TEO LT, AB teikiamos paslaugos
Adresas	Kliento adresas, kur jam yra įrengta paslauga
Incidentas	Registruotas incidento faktas, apie paslaugos gedimą tam tikram klientui

2.6.4. CFIA įvertinimas pagal sistemos architektūrą

Pagal paprasčiausią CFIA šabloną, norint įvertinti kiekvieno komponento įtaką, reikia turėti lentelę, kurioje išrašyti komponentai ir sistemos teikiamos paslaugos. Supaprastinta lentelė šiai IVS architektūrai yra pateikta lentelėje (Lentelė 8. CFIA įvertinimas.) toliau.

Lentelė 8. CFIA įvertinimas.

	Sistemos paslaugos											
Komponentai	Incidento registravimas (Internet)	Incidento registravimas (Windows)	Incidento registravimas (Intranet)	Matyti incidentus (windows)	Matyti incidentus (intranet)	Svarbūs incidentai (windows)	Svarbūs incidentai (intranet)	Incidentų sprendimo trukmė	Incidentų statistika	„Off-line“ registravimas (Win)	„Off-line“ registravimas (Office)	Klientų autentifikavimas (internet)
IVS DB (bendra)	X		X		X		X					
Lokali IVS DB[1]		X		X		X				X		X
BizTalk												
BizTalk BAM DB								X	X			
Windows App		X								X		
Office InfoPath											X	
Internet App	X											X
Internet User DB	X											X
MS Sharepoint											X[2]	
Intranet App			X		X							

Kad windows programa būtų kuo mažiau priklausoma nuo išorinių faktorių, tokių kaip tinklo sutrikimai, centrinės DB gedimai ir panašiai – siūloma kiekvienam windows klientui saugoti pačią būtiniausią informaciją lokaliai, tame pačiame kompiuteryje kaip ir programa. Tai galima įgyvendinti naudojant Microsoft SQL Server 2005 Express Edition duomenų saugojimui. Tokiu atveju, vartotojai negalėtų registruoti gedimo (svarbiausias reikalavimas) tik tuo atveju, jei neveikia pati programa arba

[1] Kiekviena veikianti programinės įrangos kopija turi savo lokalią DB, dėl to pasiekiamumas lygiagretinasi. Pvz.: Jei turime 20 darbo vietų su IVS windows programa – vadinasi sistemoje bus 20 veikiančių IVS lokalių DB windows programoms.

[2] Vartotojas gali saugoti dokumentą savo kompiuteryje, tačiau neveikiant MS Sharepoint sistemai, jis negalės perduoti incidento informacijos į sistemą. Tai yra problema tik tiek, kiek tai trunka daugiau, nei SLA trukmė.

vietinė duomenų bazė, tačiau pastarųjų mes galime turėti labai daug, t.y. praktiškai tiek – kiek yra darbo vietų. Savo ruožtu tai reiškia, jog sistema yra išlygiagretinama.

Tam, kad įvykus kritinei situacijai incidentas užregistruotas viename kompiuteryje, būtų matomas kitame (kai nėra prieinama centrinė bazė) – reikia šia informacija dalintis. Vienas iš galimų būdų tai atlikti – naudotis „Peer Network“ (arba tiesiog P2P) protokolu ir registruojant incidentą lokaliai šią informaciją pranešti visiems kompiuteriams dalyvaujantiems incidentų registravimo procese.

Dar vienas būdas registruoti incidentus „off-line“ režime, jeigu neveikia niekas (įskaitant ir lokalią DB) – tai naudotis Office Infopath formomis. Pastarosios gali būti sukonstruotos taip, kad nereikalautų centrinės duomenų bazės. Tai leistų vartotojams įvesti informaciją net tada, kai kompiuteris yra ne tinkle ir nėra lokaliai suinstaliuota jokia duomenų bazė. Tokį sprendimą klerkai gali naudoti ir kaip atsarginį incidentams registruoti, jei neveikia pagrindinė programa.

Naudojantis Infopath formomis incidentai išsaugomi kompiuteryje ir vėliau gali būti įrašyti į sistemą išsaugotus dokumentus tiesiog perkeliant į atitinkamą Sharepoint biblioteką tada, kai ji yra pasiekiama.

2.7. Detali sistemos architektūra

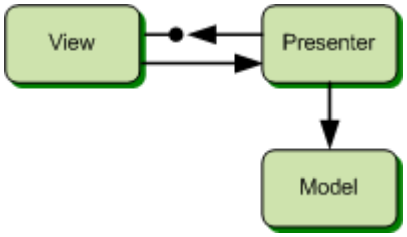
Sistema bus kuriama naudojant Microsoft BizTalk Server, Visual Studio 2005 ir .Net framework priemones.

2.7.1. Vartotojo sąsajos

Šioje sistemoje numatytos trys vartotojo sąsajos:

- Windows programa klerkams;
- Web programa naudojama intranete;
- Web programa, klientų naudojama iš interneto;

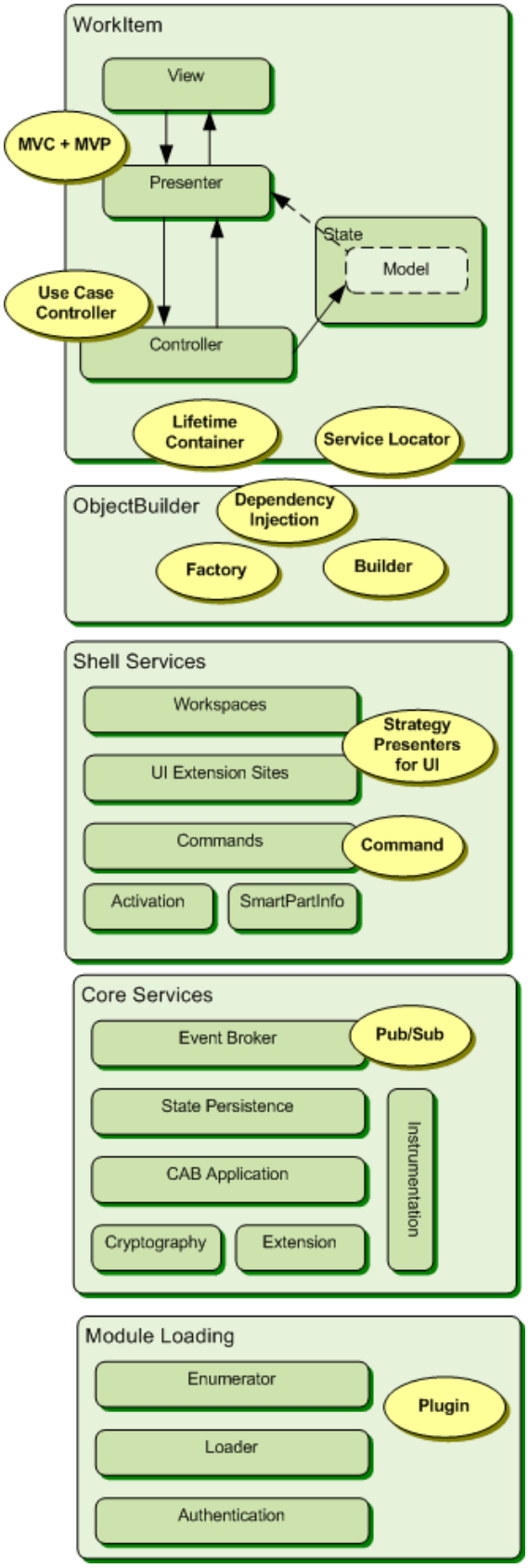
Visos šios sąsajos padarytos remiantis MVP šablonu (Paveikslas 11).



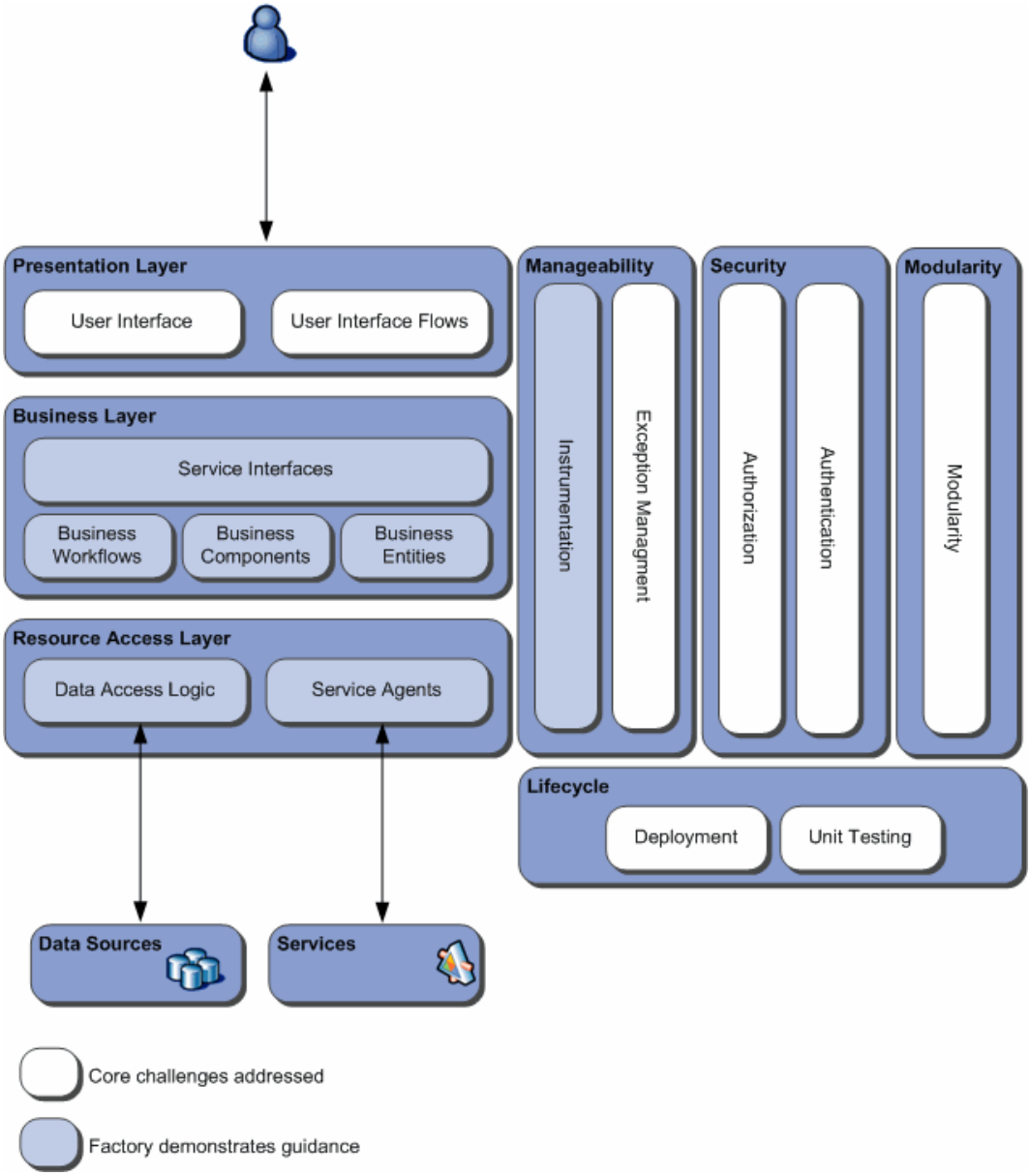
Paveikslas 11. MVP šablonas.

Windows programa bus sukurta naudojant SCSF (6). Principinė SCSF schema pateikta tolliau (Paveikslas 12). Web programai kurti bus naudojamas WCSF (7), kurios principinė schema pateikta

paveikslėlyje (Paveikslas 13) žemiau. Tiek Windows, tiek web programų atveju sistemą sudarys vaizdai (angl. „View“), kuriais vartotojas iš esmės ir naudosis. Iš esmės skirsis tik vaizdams realizuoti naudojamos technologijos. Windows atveju – tai Windows Forms, Web – ASP.Net ir ASP.Net AJAX.



Paveikslas 12. SCSF principinė schema (6).



Paveikslas 13. WCSF principinė schema (7).

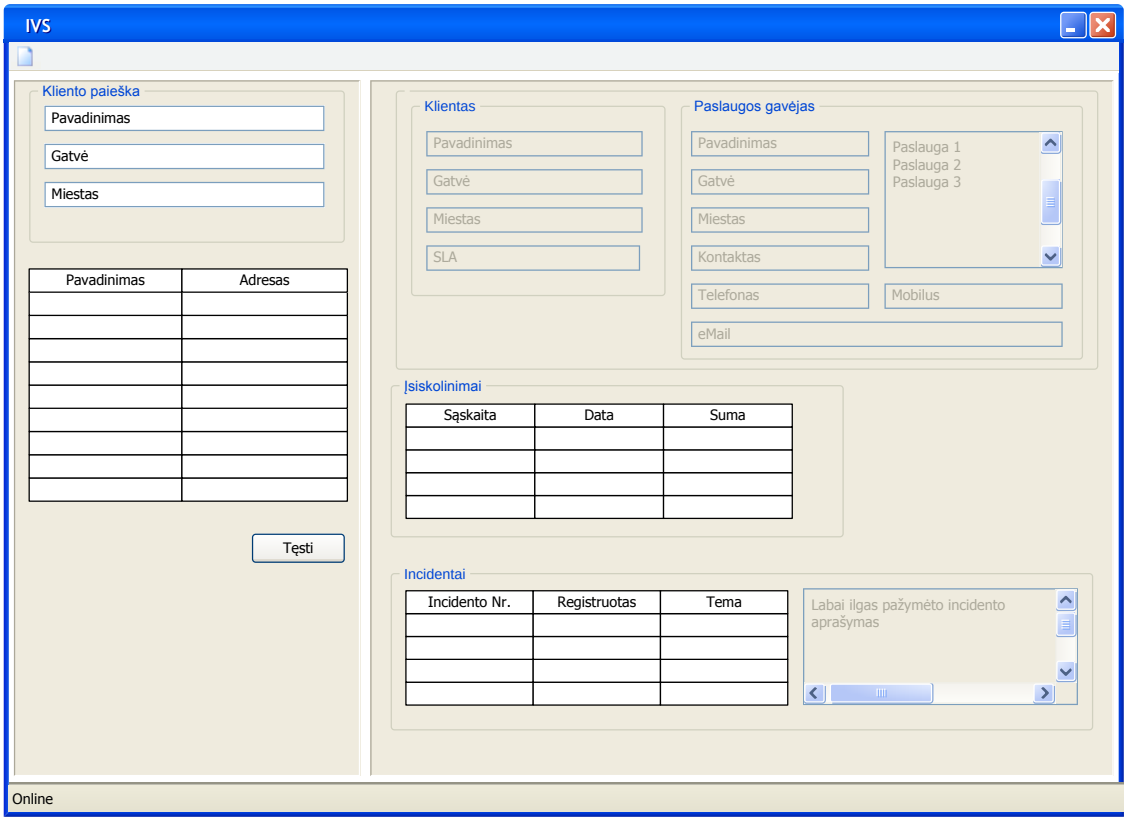
Sistemos ataskaitos reikalingos reikalavimams patenkinti bus realizuotos naudojant Microsoft SQL Server Reporting Services (SSRS), bei pasiekiamos per SSRS standartinį puslapį.

2.7.1.1. Bendri vaizdai

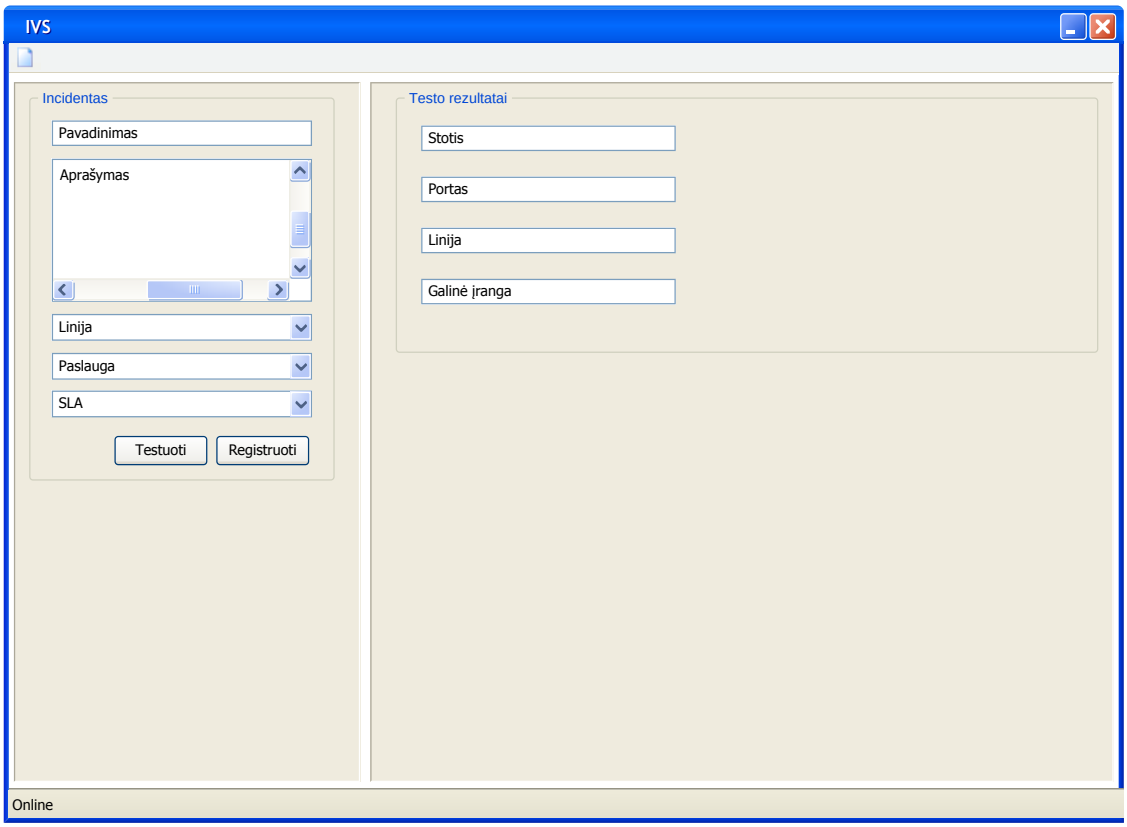
Pagal reikalavimus ir panaudos atvejus yra identifikuoti šie langai:

- Kliento ir su juo susijusios informacijos paieškos
- Incidento registravimo
- Incidentų peržiūros

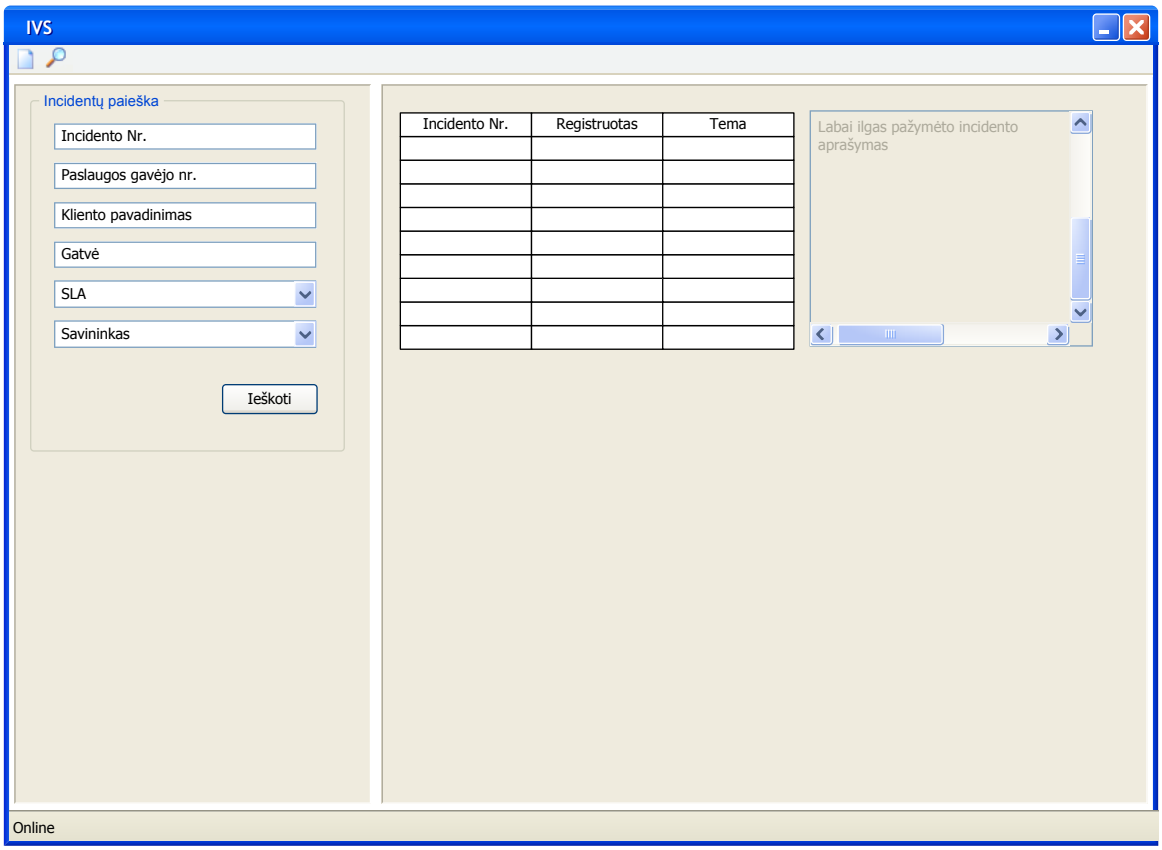
Principiniai langų vaizdai pavaizduoti paveiksluose žemiau.



Paveikslas 14. Kliento informacijos paieškos langas.



Paveikslas 15. Incidento registravimo langas.



Paveikslas 16. Incidento paieškos langas.

Iš šių ekranų identifikuoti vaizdus, kuriuos reikėtų suprogramuoti:

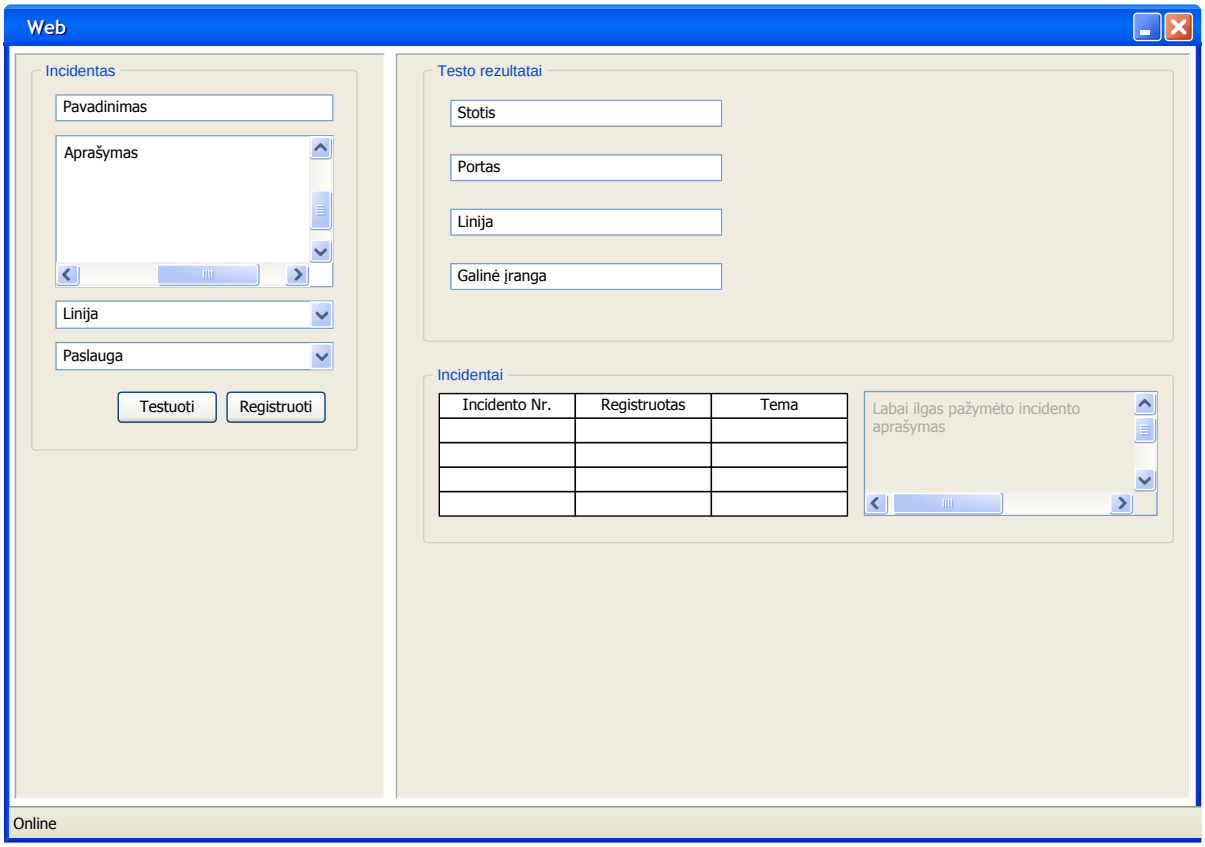
Lentelė 9. Sistemos vaizdų sąrašas.

Vaizdas	**Funkcija**
Kliento paieškos	Vartotojas gali ieškoti kliento pagal keletą tipinių parametrų
Incidento paieškos	Vartotojas gali ieškoti vieno arba keleto incidentų atitinkančių tam tikrą kriterijų
Incidento įvedimo	Vartotojas registruoja incidentą ir patikrina incidento tinklo parametrus
Kliento informacijos	Atvaizduojama kliento informacija skaitymui
Kliento įsiskolinimų	Atvaizduojama kliento įsiskolinimų informacija
Tinklo testų	Atvaizduojama informacija gauta iš tinklo testavimo posistemių
Incidentų sąrašo	Atvaizduoja registruotus incidentus skaitymui

Pateikti vaizdai yra bendri tiek Windows, tiek intranetinei programai, bei realizuojami kaip ekrano dalies vaizdai (angl. „*user control*“).

2.7.1.2. Papildomi vaizdai

Internetinei programai vaizdas šiek tiek skiriasi ir yra iš esmės paprastesnis. Taip pat, vartotojas gali matyti tik savo incidentus ir negali atlikti papildomų funkcijų (pavyzdžiui pasirinkti SLA lygio), todėl internetinei programai incidentų ekranas kiek skirsis ir ekrano pavyzdys parodytas paveikslėlyje (Paveikslas 17. Incidentų registravimo puslapio struktūra internete.) žemiau:



Paveikslas 17. Incidentų registravimo puslapio struktūra internete.

Greta aukščiau pavaizduoto lango, web puslapyje turėtų būti ir pagrindinis puslapis, autentifikavimosi puslapis, kur vartotojas gali suvesti vartotojo vardą ir slaptažodį.

2.7.1.3. Ataskaitos

Sistemoje iš esmės reikalingos dvi ataskaitos:

- Ataskaita, kurioje vaizduojami konkretūs incidentai, bei informacija susijusi su jais, tokia kaip incidento sprendimo statusas;
- Ataskaita, kurioje matyti analitinė agreatuota informacija, pvz.: incidentų trukmė pagal paslaugas ir panašiai;

Tiek vienai, tiek kitai informacijai pagrindas gali būti Microsoft BizTalk BAM duomenų bazė, kurioje gali būti išsaugoma informacija apie incidentą ir jo sprendimo eigą.

Labiausiai dominančios ataskaitos yra šios:

- Gedimų skaičius užregistruotas tam tikrose stotyse per laiko tarpą;
- Vidutinė trukmė gedimui pašalinti pagal paslaugą;
- Vidutinė trukmė gedimui pašalinti pagal vietovę;
- Gedimų skaičius laikotarpiui pagal vietovę ir datą;

Tam, kad būtų galima realizuoti numatytas ataskaitas, būtina kaupti atitinkamus faktus. Gedimo šalinimo trukmes galima įvertinti pagal BizTalk proceso automatiškai atliekamą įvykių registravimą.

Norimoms ataskaitoms sukurti bus reikalinga ši informacija:

Lentelė 10. Ataskaitų modelis.

Informacija	Tipas	Aprašymas
Trukmė	Matavimas	Matuojama trukmė nuo incidento registravimo iki momento, kai jis išsprendžiamas
Data	Dimensija	Nurodoma data, kai buvo registruotas incidentas
Vieta	Dimensija	Miestas, kuriame registruotas incidentas, bei yra nurodytas paslaugos gavėjas
Skaičius	Matavimas	Incidentų skaičius
Klientas	Dimensija	Kliento pavadinimas
Paslaugos gavėjas	Dimensija	Paslaugos gavėjo numeris (skaičius)
Paslauga	Dimensija	Paslaugos pavadinimas
Incidento svarba	Dimensija	Incidento lygio raidė (‚A‘, ‚B‘, ‚C‘, ‚D‘)
Linija	Dimensija	Linijos numeris (telefono numeris)

2.7.2. Duomenų bazės

Sistemoje numatytos dvi duomenų bazės:

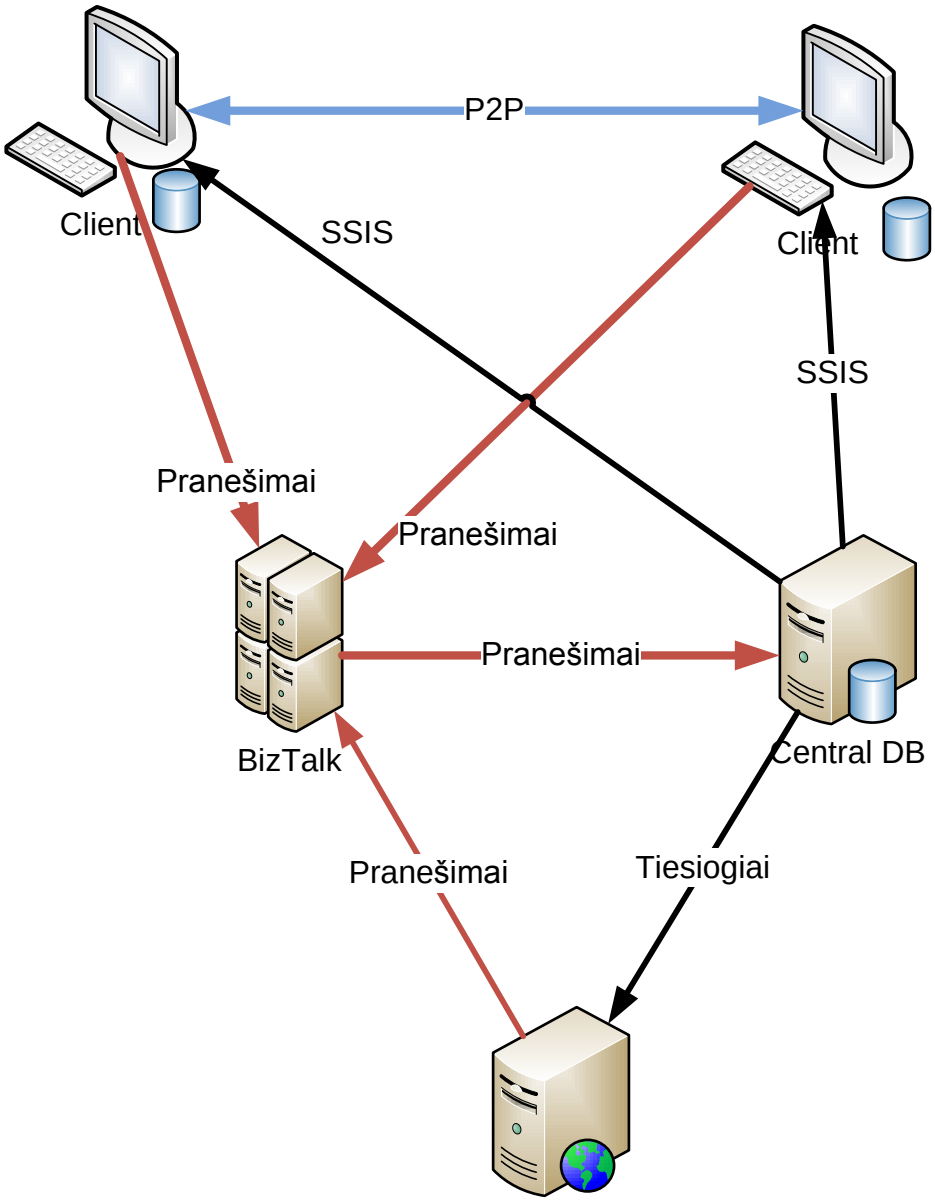
- Bendroji IVS duomenų bazė;
- Kiekvienai Windows programos instaliacijai priskirta duomenų bazė;

Kaip pagrindinis sistemos šaltinis traktuojama bendroji duomenų bazė, tačiau dėl gedimų tinkle ji gali būti neprieinama, todėl Windows programa dirba su savo baze kaip pagrindine. Kad duomenys web programose ir windows programose būtų vienodi, pastarieji turi būti sinchronizuojami.

Kad duomenys iš visų šaltinių (interneto, intraneto, ofiso programų) pakliūtų į bendrąją duomenų bazę, už jos atnaujinimą atsakomybė deleguojama BizTalk sistemai, o iš bendrosios DB informacija yra sinchronizuojama į kliento kompiuteriuose esančias Windows programų duomenų bazes.

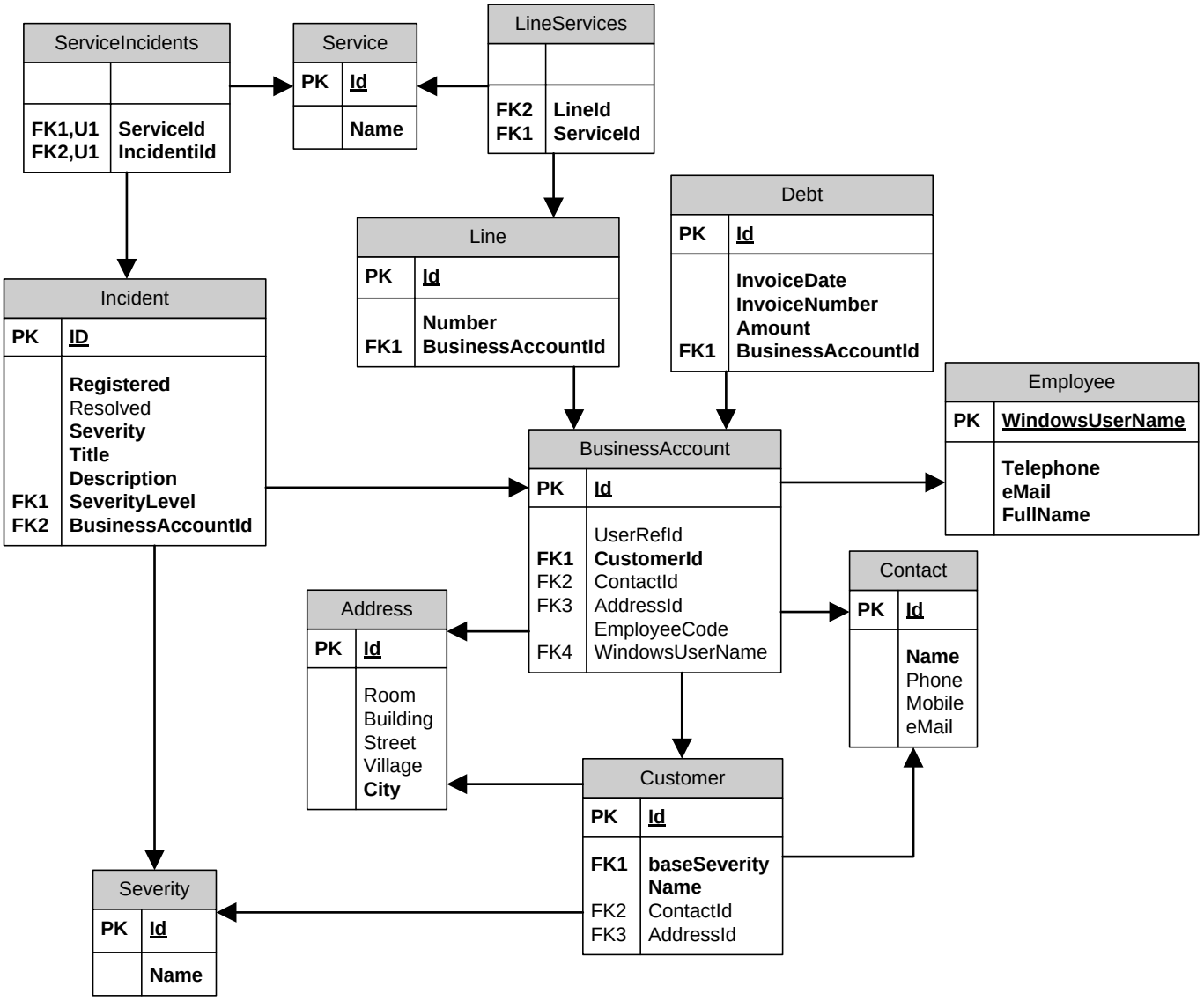
Šio informacijos perdavimo schema yra pavaizduota paveiksle (Paveikslas 18) esančiame toliau.

Pati duomenų bazės struktūra centrinėje duomenų bazėje ir kliento kompiuteriuose esančiose duomenų bazėse yra tokia pati. Skirtumas yra tik duomenų kiekyje. Kadangi diskinės talpos dydis kliento kompiuteriuose palyginti ribotas, dėl to nuspręsta saugoti tik santykinai statinę klientų, tinklo, skolų informacija, bei einamuosius duomenis ir savaitės incidentų archyvą. Tuo tarpu centrinėje duomenų bazėje galima kaupti incidentų archyvą didesniam laikotarpiui.



Paveikslas 18. Duomenų sinchronizacimos schema.

Žemiau pateikta diegiamos duomenų bazės schema. Informacija apie klientus, paslaugų gavėjus, linijas ir pan. yra pakraunama iš kitų sistemų. IVS vartotojo sąsajoje registruojami tik nauji incidentai – t.y. pildoma Incident lentelė.



Paveikslas 19. Duomenų bazės schema.

2.7.3. Sistemos klasės ir komponentai

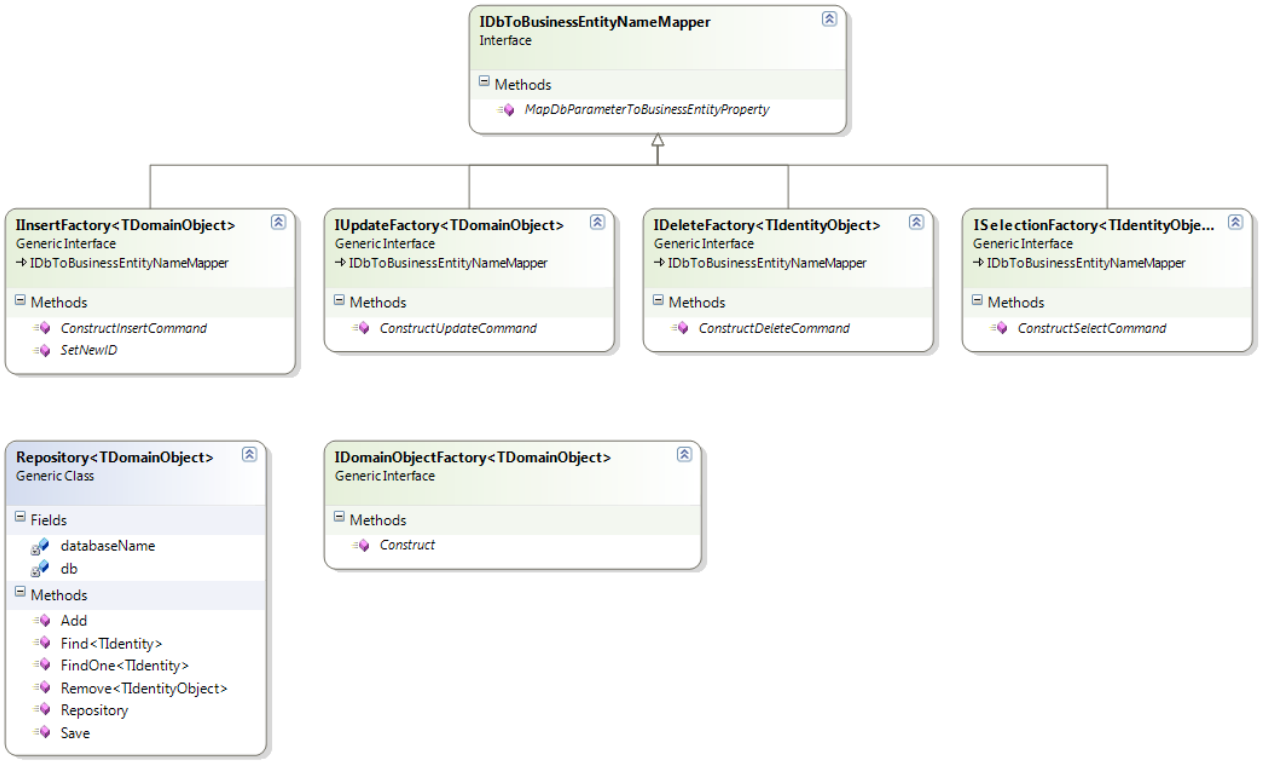
Kiekvienos programos komponentus galima logiškai sugrupuoti į tam tikrus lygius (8):

- Vartotojo sąsajos;
- Verslo logikos;
- Duomenų prieigos;
- Duomenų bazių ir paslaugų;

Vartotojo sąsajos komponentai aptarti paragrafe 2.7.1.

2.7.3.1. Duomenų prieigos komponentai

Duomenų skaitymui-rašymui bus naudojama Microsoft .Net SqlDataReader klasė, bei tipinis tokiais atvejais taikomas šablonas pavaizduotas paveikslėlyje.



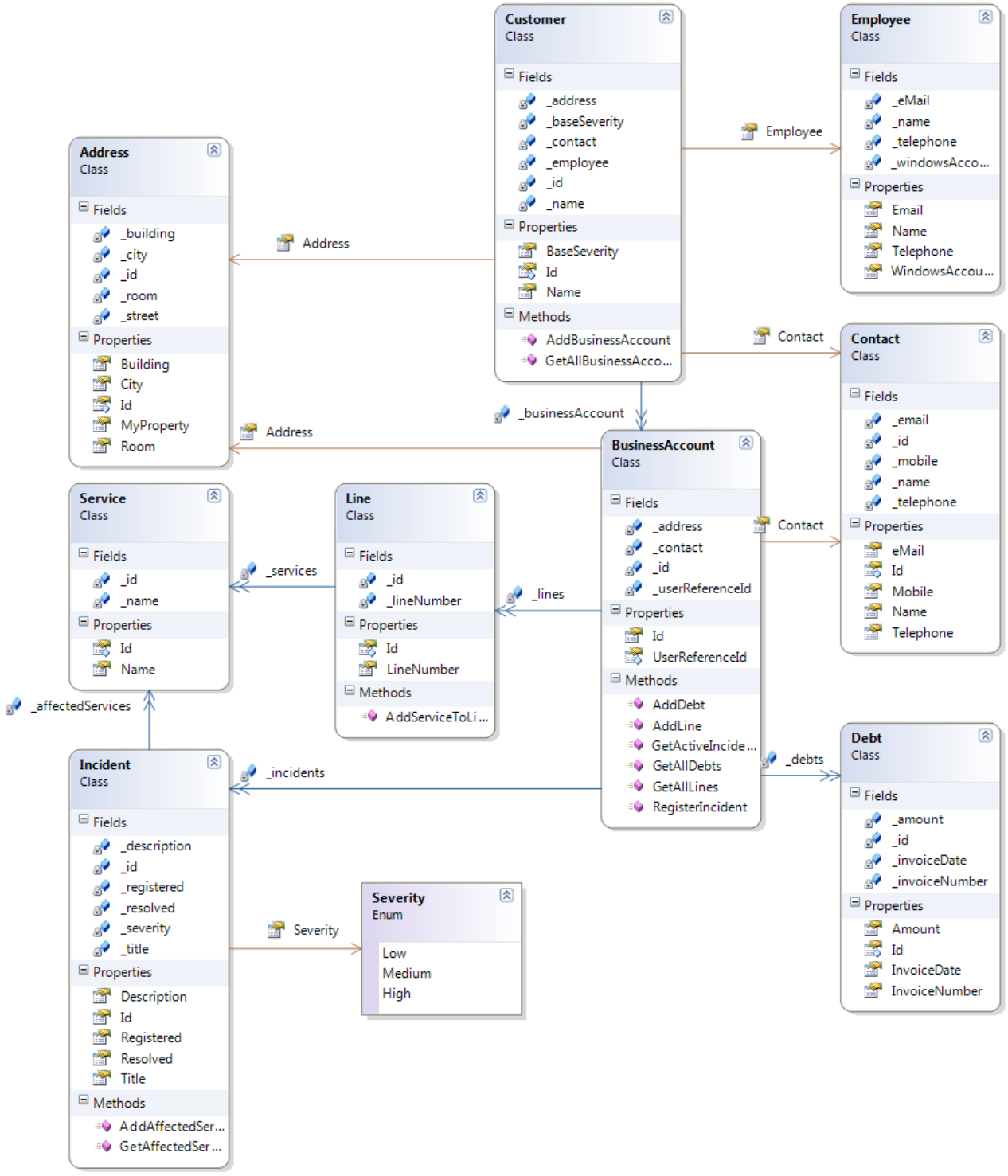
Paveikslas 20. Duomenų prieigos šablonas.

Lentelė 11. Pagrindinės duomenų prieigos klasės.

Tipas	Aprašymas
IDomainObjectFactory	Šio interfeiso klasė susieja SqlDataReader nuskaitytą eilutę su objektu
Repository	Bazinė klasė operacijoms su objektais
IDBToBusinessEntityMapper	Realizuojanti klasė susieja išsaugotos procedūros parametrą su verslo objektu
IInsertFactory, IUpdateFactory, IDeleteFactory, ISelectionFactory	Realizuojančios klasės turi užtikrinti atitinkamų komandų sukūrimą

2.7.3.2. Sistemos esybės

Sistemoje naudojamų verslo esybių klasių diagrama pateikta paveikslėlyje (**Error! Reference source not found.**) žemiau. Minėtoje diagramoje atvaizduotos tik klasės susijusios su pačia sistema. Microsoft .Net Framework 2.0 infrastruktūroje esančios klasės (tokios kaip User) šioje diagramoje neatvaizduotos.



Paveikslas 21. Sistemos esybių klasių diagrama.

2.7.3.3. Verslo logika

Pagal reikalavimus identifikuojamos du verslo lygmens komponentai:

- Verslo logikos komponentas reikalingas teisingam incidentų įvedimui užtikrinti;
- Verslo procesas, būtinas kontroliuoti incidento sprendimą laike, bei integruoti aplinkines sistemas;

Kadangi didžioji dalis esybių yra kitų sistemų esybių kopijos ir jos iš esmės nėra regaduojamos, o vienintelis objektas, kuriuo operuojama – yra incidentas, dėl to nuspręsta taisykles įkelti į esybę ir realizuoti kaip klasės metodus.

Incidento registravimas atliekamas vykdant metodą RegisterIncident, BusinessAccount klasėje (Paveikslas 21), kuriame ir turėtų būti papildomai realizuotas patikrinimas:

- Ar klientas turi užsakęs paslaugas, kuriomis skundžiasi?
- Ar klientas nėra skolingas?
- Ar įvestas SLA lygis yra ne mažesnis nei priskiriamas tam konkrečiam klientui?

Verslo proceso schema iš esmės yra atspindėta paveiksle esančiame aukščiau (Paveikslas 9). Norint pasiekti, kad informacijos atnaujinimo ir kontrolės procesai būtų nepriklausomi, reikėtų vadovautis geriausiomis integravimo praktikomis naudojant BizTalk Server (9):

- Gautas incidentas iškart publikuojamas į centrinę IVS DB ir CRM sistemas – pagal publikavimo-prenumeratos šabloną (10);
- Kadangi sistemose naudojami formatai skiriasi, pranešimai yra transformuojami, t.y. naudojamas transformavimo šablonas (10);
- Verslo procesas prenumeruoja publikuotus naujus incidentus (angl. „*publish-subscribe*“);
- Baigus spręsti incidentą, BizTalk procesui siunčiamas atitinkamas pranešimas, bei naudojama pranešimų koreliacija;
- Gautas pranešimas yra paskiriamas visiems prenumeratoriams: BizTalk procesui (naudojama koreliacija), CRM bei centrinei IVS DB.

Minėtas algoritmas užtikrina, jog procesas gali būti versijuojamas atskirai, o pranešimai gali būti perduoti CRM ir centrinei IVS DB, kai tik yra tokia galimybė ir yra pakankamai resursų.

2.8. Sistemos kokybės įvertinimas

Siekiant įvertinti kuriamos sistemos kokybę būtina iš anksto numatyti tam tikrus testus, kurie atspindėtų keliamus funkcionalumo bei kokybinius reikalavimus. Pagrindiniai reikalavimai yra susiję su galimybėmis registruoti bei matyti incidentus, bei atlaikyti situacijas, kai neveikia viena ar daugiau galinių sistemų.

Kadangi atskirai instaliuotis visas galines sistemas (CRM, apskaitos, NIS, ir t.t.) yra neracionalu – pastarąsias galima pakeisti Mock objektais (11) arba servisais, kurie imituoja jų atsakymus ir kuriuos galima testavimo tikslais sustabdyti.

Žemiau lentelėje pateikti siūlomi būtini testai, kurie atspindėtų reikalavimus:

Lentelė 12. Taikytini testų scenarijai.

Testas	Aprašymas	Tipas
Incidento registravimas	Naujas incidentas registruojamas iš Windows formos, incidentas užregistruojamas lokalioje DB	Automatinis
Incidento perdavimas	Veikiant dviems ar daugiau Windows klientų viename registruojamas incidentas, kituose jis taip pat užregistruojamas.	Automatinis
Sistema veikia sustabdžius visas galines sistemas	Sustabdomas Biztalk ir verslo sistemas imituojantys servisai. Incidentus galima užregistruoti su windows klientu ir Office programa	Rankinis
Sistema paima susikaupusius pranešimus	Paleidžiami BizTalk servisai – susikaupę pranešimai yra perskaitomi	Rankinis
Registruojant incidentą su tuo pačiu identifikatoriumi jis neregistruojamas	Du kartus bandant registruoti incidentą su tuo pačiu identifikatoriumi, naujas įrašas į DB yra nerašomas	Automatinis
Paskelbus incidentą automatiškai atnaujinamos prieinamos sistemos	Windows klientu registruojamas incidentas, BizTalk ir IVS centrinė DB veikia, tačiau neveikia CRM. Incidentai registruojami IVS DB, tačiau kaupiasi pranešimai į CRM.	Rankinis

Windows klientas periodiškai pasiima informaciją iš centrinės IVS DB	Kliento starto metu ir kas 10 minučių yra atnaujinama windows kliento DB iš centrinės DB.	Automatinis
Pranešimai apie išspręstus incidentus baigia BizTalk procesą ir atnaujina CRM, bei centrinės IVS DB įrašų būklę.	Jeigu toks incidentas (su tokiu identifikatoriumi) yra užregistruotas – tai BizTalk procesas yra baigiamas ir pakeičiama atitinkamų įrašų būsena CRM ir centrinėje IVS DB. Jeigu tokio incidento nėra – sistema tokį pranešimą ignoruoja.	Rankinis

Sistemos testams, bei jų rezultatams registruoti galima naudoti Visual Studio Team Tester Edition arba panašius, bet nemokamus produktus kaip NUnit.

IŠVADOS

1. Išnagrinėjus galimus integravimo variantus parodyta, kad teisingai taikant SOA principus galima pagerinti sistemos pasiekiamumą ir našumą net esant tom pačiom infrastruktūros salygom;
2. Analizuojant reikalavimus patartina naudoti detalesnį kategorijų sąrašą (priežiūros, saugos, pasiekiamumo, ir t.t.), idant būtume tikri, jog nepraleisti nefunkciniai reikalavimai;
3. Nors šiuolaikinės priemonės ir leidžia pakankamai paprastai realizuoti sudėtingus scenarijus (tokius kaip P2P tinklai), tačiau bet kokiu atveju – tai yra papildomos išlaidos. Kitais žodžiais tariant, kiekvienas kokybinis (nekeičiantis sistemos funkcionalumo) sistemos pagerinimas kainuoja ir turėtų būti vertinamas kaip ir bet kuris funkcinis, t.y. įvertinant jų teikiamą naudą ir kaštus;
4. Kuriant sistemas, kurioms taikomi dideli pasiekiamumo reikalavimai, galima iš anksto taikyti pasiekiamumo matavimo metodologijas ir tuo būdu tikslinti sistemos architektūrą;
5. Sistemos patikimumas iš esmės priklauso nuo visų joje naudojamų komponentų patikimumo.
6. Kai komponentai yra išoriniai įmonės atžvilgiu (kai sistemos parametrai nėra kontroliuojami tos pačios organizacijos), reikia įvertinti ir tokius galimus variantus, kai komponentų našumo parametrai ims netenkinti reikalavimų. Pvz.: iki tam tikro momento mokesčių inspekcijos web paslaugomis naudojasi tik viena įmonė ir našumas yra pakankamas, tačiau prisijungus daug daugiau vartotojų esamų resursų gali pradėti nepakakti;
7. Testų rinkinį galima panaudoti kaip priemonę reikalavimams tikslinti, bei galimai vėliau tikrinti sistemos veikimą jau produkcijos aplinkoje;

LITERATŪRA

1. **net.com.** Advanced Fault Management System. *Advanced Fault Management System.* [Tinkle] https://internet.net.com/products/nms/afms.shtml.

2. **Manage Engine.** ManageEngine ServiceDesk Plus 6. *Manage Engine Web site.* [Tinkle] http://manageengine.adventnet.com/products/service-desk/index.html.

3. **Bartlett, John, et al.** *Service Delivery.* s.l. : OGC by TSO, 2001. 978-0113300174 .

4. Wikipedia. *Availability.* [Tinkle] 2007 m. http://en.wikipedia.org/wiki/Availability.

5. **Trowbridge, David, et al.** Integration Patterns. *MSDN Library.* [Tinkle] http://msdn2.microsoft.com/en-us/library/ms978729.aspx.

6. **Microsoft.** Smart Client Software Factory. *MSDN.* [Tinkle] 2006 m. June. http://msdn2.microsoft.com/en-us/library/aa480482.aspx.

7. —. Web Client Software Factory. *MSDN.* [Tinkle] 2007 m. January. http://msdn2.microsoft.com/en-us/library/bb264518.aspx.

8. —. Application Architecture for .NET: Designing Applications and Services. *MSDN.* [Tinkle] 2002 m. http://msdn2.microsoft.com/en-us/library/ms954595.aspx.

9. **Wasznicky, Marty ir Zimmerman, Scott.** 8 Tips And Tricks For Better BizTalk Programming. *MSDN Magazine.* [Tinkle] Microsoft Corporation and CMP Media, LLC, 2007 m. http://msdn.microsoft.com/msdnmag/issues/07/05/BizTalk/Default.aspx.

10. **Hohpe, Gregor ir Woolf, Bobby.** *Enterprise Integration Patterns.* Boston : Addison-Wesley, 2004. 0-321-20068-3.

11. Mock object. *Wikipedia.* [Tinkle] 2007 m. http://en.wikipedia.org/wiki/Mock_Object.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

NIS	Tinklo informacinė sistema. Sistema, kurioje apskaitomi tinklo elementai.
CRM	Ryšių su klientais valdymo sistema.
NMS	Tinklo valdymo sistema, atliekanti aktyvų tinklo stebėjimą ir tam tikrų tinklo elementų valdymą.
WFM	Darbo resursų valdymo sistema, skirta gedimo vietoje dirbančių inžinierių darbo laiko planavimui
HD	„Pagalbos stalo“ (angl. *„Help desk“*) sistema
MTBF	Vidutinis laikas tarp gedimų
MTTR	Vidutinis laikas reikalingas gedimui pašalinti
MTBSI	Vidutinis laikas tarp incidentų
SOA	Orientuota į paslaugas architektūra
SLA	Paslaugų lygio sutartis
IS	Kompiuterinė informacinė sistema
OS	Operacinė sistema
RFI	Pradinė tiekėjų apklausa
IVS	Incidentų valdymo sistema (atliekamas projektas)
CFIA	Komponento gedimo įtakos analizė
MVP	Angl. *„Model-View-Presenter“* – šablonas taikomas vartotojų sąsajoms, bei funkcionalumui atskirti.
SSRS	Microsoft SQL serverio ataskaitų paslaugos

1 PRIEDAS. Pradiniai reikalavimai

Šiame priede pateikti iš RFI dokumento išgryninti reikalavimai surašyti į Excel failą.

Lentelė 13. Pradiniai reikalavimai

Nr.	Reikalavimas	Paaiškinimas
1	Darbuotojas gali užregistruoti incidentą	
2	Darbuotojas prieš užregistruodamas turi patikrinti ar klientas neskolingas	Jeigu klientas skolingas, jam paslauga galėjo būti išjungta
3	Darbuotojas patikrina koks tinklo elementas gali neveikti	
4	Jeigu paslauga leidžia, patikrinti paslaugos veikimą iškart	DSL paslaugai galima patikrinti liniją
5	Darbuotojas patikrina kliento pateiktą informaciją	
6	Sistema turi būti realizuota kaip web programa	
7	Darbuotojas gali naršyti po registruotus incidentus	
8	Darbuotojas gali ieškoti užregistruoto incidento	
9	Užregistruoto incidento informacija, kartu su kontaktine ir tinklo informacija turi būti perduota į WFM	Toliau darbuotojų resursų valdymas vyksta WFM programoje
10	Skambučių centrų darbuotojai informuoja klientą apie pašalintą gedimą ir klausia patvirtinimo	

2 PRIEDAS. Reikalavimai programinei įrangai

Reikalavimai
Programinė įranga sukurta .Net framework 1.1/2.0/3.0 versijos pagrindu
Programinės įrangos ar jos dalių instaliacija yra: • Windows Installer 3.0 formate (MSI failai) • Palaiko instaliacijos režimą be vartotojo įsikišimo („silent“) • Nereikalauja vartotojo su administratoriaus teisėmis Arba • Yra sukurta .Net pagrindu ir įsidiegia automatiškai paleidus Internet Exploreryje be administratoriaus teisių
Programinė įranga palaiko automatinį instaliavimą, atnaujinimą bei pašalinimą naudojant Windows 2003 Active Directory grupių politikas
Programinė įranga veikia su Windows XP ir Windows Vista vartotojo (User) teisėmis
Jei pateikiamas programinės įrangos kodas, įranga sukurta: • naudojant Visual Studio.Net 2003, 2005 versijas • Visual Basic.Net arba C# kalbomis. • Programinė įranga, sukurta pagal UŽSAKOVO užsakymą, tampa jo nuosavybe ir įgalina pilnai disponuoti (keisti, naudotis) išeities tekstais.

Klientinė programinė įranga veikia Windows XP Professional arba Windows Vista Enterprise aplinkose ir turi būti suderinta su standartinės darbo vietos įranga: • Microsoft Office 2003/2007 • Internet Explorer 7.0 • Acrobat Reader 7.0 • Virus Scan TC 6.1 • NetMeeting 3.01 • Alkonas • .Net framework 1.1, 2.0, 3.0 • ABBYY Form Filler 2.0 Light Klientinė programinė įranga turi palaikyti standartinius nustatymus: • Laiko zona - (GMT +02:00) Tallinn, Riga, Vilnius su perėjimu į vasaros/žiemos laiką: o Standartinės datos yra užrašomos YYYY.MM.DD, kur YYYY - metai, MM – mėnuo ir DD – diena; o Tūkstančiai atskiriami kableliais, o dešimtosios dalys – tašku. • Standartinė rekomenduojama rezoliucija - 1280x1024x16bit 85Hz. • Pašto žinutėms naudojamas Baltic (Windows) standartas.
Klientinė programinė įranga palaiko ilgus ir atitinkančius Unified Naming Convention (UNC) failų, bei spausdintuvų vardus
Programinė įranga netalpina ir nekeičia duomenų Windows operacinės sisteminiuose kataloguose
Techninės įrangos tvarkyklės (drivers) yra WHQL sertifikuotos
Vartotojams autorizuoti sistema naudojasi: • Windows autorizacija • palaiko „single sign-on“ • naudojasi Windows domeno vartotojo ar grupių teisėmis, įskaitant duomenų bazių serverius
Sistema archyvuoja duomenis, jeigu duomenys yra nuolatos papildomi
Palaikoma serverinė įranga: • Windows 2003

Duomenų bazių serveriai: • SQL Server 2000 32/64 bit • SQL Server 2005 32/64 bit
Integravimui su kitomis programomis naudojami „XML Web Services“
Elektroniniam paštui naudojamas Exchange 2003
Web turiniui atvaizduoti naudojamas: • Windows 2003 Internet Information Server 6.0
Integravimo infrastruktūra – BizTalk Server 2006
Palaikomas išėjimas į internetą per ISA Server 2004
Programinė įranga atitinka įmonėje naudojamų tinklų reikalavimus: • 10/100 Mbit/s Ethernet LAN. • WAN – min. 2Mbit/s. • Ryšio protokolas – TCP/IP.

3 PRIEDAS. Papildyti reikalavimai sistemai

Lentelė 14. Patikslinti reikalavimai

Nr.	Reikalavimas	Šaltinis	Prioritetas	Pastabos
1.	Vartotojas gali užregistruoti gedimą sistemoje	Gintarė	1	
1.1	Jei gedimas klientui užregistruotas, pakartotinai neregistruoti	Gintarė	1	
1.2	Jeigu klientas įsiskolinęs, gedimo neregistruoti	Gintarė	1	
1.2.1	Jeigu skolininkų informacija nepasiekiama, gedimą registruoti	Gintarė	2	
1.2.2	Įsiskolinimo informaciją būtina patikrinti	Gintarė	2	
1.3	Incidentus gali norėti pamatyti, bei užregistruoti „atsitiktiniai“ vartotojai	Gintarė	2	Kartais informacija apie incidentą gali pasiekti ne klerką, bet kitą įmonės darbuotoją ir jis turėtų turėti galimybę užregistruoti incidentą
1.4	Proceso dalyviai turi matyti aukšto prioriteto incidentus	Gražvydas	1	
1.4.1	Incidentų prioritetai nustatomi pagal kliento informaciją	Gražvydas	1	Registruojantys darbuotojai gali atpažinti tokius klientus ir esant situacijai, kai CRM sistema nepasiekiama
1.4.2	Priskirti vadybininkai turi matyti užregistruotus incidentus	Gintarė	1	Šie žmonės atsakingi už svarbių incidentų sprendimo kontrolę
1.5	Incidentų sąrašas turi būti pasiekiamas proceso vadovui	Gintarė	2	
1.5.1	Incidentai, kurie sprendžiami ilgiau nei SLA numatytas laikas, turi būti parodyti atskirai	Gintarė	2	
2.	Klientas gali užregistruoti gedimą	Vytautas	2	
2.1	Gedimą galima registruoti įmonės interneto svetainėje	Vytautas	2	
2.2	Gedimą galima registruoti siunčiant el. laišką nurodytu adresu	Vytautas	2	
3.	Gedimų šalinimo vadovui turi būti pateikti incidentų statistika per paskutinę parą	Gražvydas	2	Reikalinga, norint identifikuoti masinius gedimus

4.	Proceso vadovams turi būti pateikta ataskaita apie gedimų šalinimo eigą	Vytautas	2	Reikalinga, norint įvertinti proceso, darbuotojų našumą
5.	Sistemoje turi būti numatytas scenarijus, kaip registruoti gedimus, kai nepasiekiamos sistemos dalys	Gintarė	1	
5.1	Visada turi būti pasiekiami aukšto lygio incidentai	Gražvydas	1	
5.2	Sistema turi veikti ir leisti užregistruoti incidentą, jei nepasiekiama kuri nors integruojama IS	Gražvydas	1	
5.3	Turi būti numatytas "off-line" incidentų registravimo scenarijus	Gintarė	1	
5.3.1	Sistema turi leisti užregistruoti incidentą nesant ofise	Gražvydas	2	
5.3.2	Sistema turi leisti užregistruoti incidentą tinkle, jei neprieinamas duomenų centras	Almantas	1	
5.3.3	Sistemoje neturėtų reikėti „popierinio“ varianto	Gintarė	1	
6.	Sistema turi pateikti priežiūrai reikalingą informaciją	Almantas	2	
6.1	Sistema turi identifikuoti neveikiančius komponentus	Almantas	2	
6.1.1	Informacija apie neveikiančius komponentus turi būti registruojama įvykių žurnale		2	
6.2	Sistema neturi rodyti techninių klaidų pranešimų neveikiant sistemos komponentams	Almantas	1	
6.3	Administratorius turi būti informuojamas esant gedimams	Almantas	2	

4 PRIEDAS. Priimti architektūriniai sprendimai.

Lentelė 15. Architektūriniai sprendimai.

Sprendimas	Pagrindimas
Klerkams Windows programa	Ši programa gali veikti „off-line“ režime.
Windows programa turi turėti lokalią DB	Kad sistemos veikimas minimaliai priklausytų nuo išorinių resursų – būtini duomenys turi būti prieinami lokaliai.
Net Peer (P2P) protokolas windows programų	Tokiu būdu galima užtikrinti informacijos pasidalinimą, kai neveikia serverinė dalis. Taip minimizuojama priklausomybė nuo serverių infrastruktūros, bei maksimaliai padidinamas klientų skaičius.
Asinchroninis registravimas	Incidentai sistemose turi būti registruojami asinchroniškai, kad servisai galėtų reguliuoti savo apkrovą, bei užtikrinti informacijos perdavimą, jei galinė sistema nepasiekiama.
Office InfoPath šablonas incidentų registravimui	Vadybininkas ar kitas asmuo gali užpildyti informaciją apie incidentą/problemą pas klientą, išsaugoti informaciją, o vėliau persiųsti į sistemą.
Internetinis puslapis	Internetinis puslapis klientams registruoti savo gedimus. Jo prieinamumas priklausys nuo serverio būvio;
Intranetinis puslapis	Puslapis atsitiktiniams vartotojams skirtas registruoti bei peržiūrėti incidentus. Nereikalauja diegimo, tačiau priklauso nuo web serverio pasiekiamumo.
Incidento BizTalk procesas (-ai)	Užtikrinti gedimo registravimą, proceso kontrolę ir ataskaitas.
Tiesioginė pranešimų prenumerata	Nepriklausomos sistemos turėtų „prenumeruoti“ pranešimus tiesiogiai, nes kontroliuoti pranešimų persiuntimą procese nėra prasmės.