



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Joel Moisio, Miro Salo

HELPDESK-KÄYTTÖLIITTYMÄN RAKENTAMINEN

Case Lemonsoft Oy

Liiketalous
2016

TIIVISTELMÄ

Tekijä	Joel Moisio, Miro Salo
Opinnäytetyön nimi	Helpdesk-käyttöliittymän rakentaminen
Vuosi	2016
Kieli	suomi
Sivumäärä	48
Ohjaaja	Raija Tuomaala

Tämä opinnäytetyö käsittelee helpdesk-käyttöliittymäratkaisua nykyaikaisilla teknologioilla toteutettuna. Työn tavoitteena oli rakentaa uusi käyttöliittymä helpdesk-järjestelmälle. Työn toimeksiantaja oli Lemonsoft Oy.

Käyttöliittymä toimisi osana Lemonsoft Oy:n onlinesovellusta ja tulisi yrityksen työntekijöiden käyttöön. Käyttöliittymän kautta kirjattaisiin ja käsiteltäisiin yritykselle saapuvia tukipyyntöjä. Käymme työssä läpi keskeisimmät teknologiat sekä ohjelmistokehykset, joita työn toteutuksessa käytettiin. Työssä esitellään pääasiassa AngularJS-ohjelmistokehyksen ominaisuuksia.

Työn tuloksena syntyi toimiva helpdesk-järjestelmän käyttöliittymä, joka on Lemonsoft Oy:llä käytössä. Käyttöliittymä on saanut paljon positiivista palautetta ja sen on sanottu olevan edellistä käyttöliittymää parempi ja sujuvampi. Lähitulevaisuudessa käyttöliittymä korvaa kokonaan edeltäjänsä ja Silverlight-alustalla toimiva käyttöliittymä poistetaan käytöstä.

ABSTRACT

Author	Joel Moisio, Miro Salo
Title	Creating a User Interface for Helpdesk Use
Year	2016
Language	Finnish
Pages	48
Name of Supervisor	Raija Tuomaala

This thesis studied a helpdesk interface solution implemented with modern technologies. The aim was to build a new interface for a helpdesk system. The study was commissioned by Lemonsoft Oy.

The helpdesk interface would work as a part of Lemonsoft Oy's online product. The user interface would be used by the company's employees. The interface would be used to handle and create the feedbacks received by the company. In this the most important technologies and frameworks used in the implementation of the interface were examined. This thesis focuses mainly on AngularJS software framework features.

The result was a working helpdesk system interface, which the client is now using. The user interface has received a lot of positive feedback, and it is said to work better and more fluid than the previous interface. In the near future the user interface will completely replace its predecessor, and the current interface that works on Silverlight will be disabled.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	8
	1.1 Toimeksiantaja	8
	1.2 Tavoitteet	8
	1.3 Työnjako.....	8
	1.4 Ohjelmointiteknologiat käyttöliittymän toteutuksessa	9
	1.5 Tutkimusaiheet	9
	1.6 Osaaminen	10
2	MVC.....	12
3	ANGULARJS.....	13
	3.1 Direktiivi.....	13
	3.2 Moduuli	13
	3.3 Kontrolleri.....	14
	3.4 Toistin: ng-repeat	14
	3.5 Näytä/Piilota: ng-hide ja ng-show	15
	3.6 Jos: ng-if	15
	3.7 Luokka: ng-class.....	15
	3.8 Synkroninen suoritus: \$q	15
	3.9 \$http -palvelu.....	15
4	MUUT OHJELMOINTITEKNOLOGIAT	17
	4.1 Typescript.....	17
	4.2 Automapper	17
	4.3 ASP.NET Web API.....	17
	4.4 Bootstrap.....	17
	4.5 Highcharts.....	18
5	TOTEUTUS	19
	5.1 Rakenne	19
	5.2 Tukikeskus.....	20
	5.3 Tukipyyntölista.....	26

5.4 Uusi tukipyyntö	34
5.5 Tukipyyntö	40
6 YHTEENVETO	46
LÄHTEET.....	48

KÄSITTEET

DOM

Document Object Model on lyhenne, jolla on tapa kuvata HTML:n rakennetta puuna.

ENUM

Enumerated tyypellä voidaan määrittellä teksti arvolle numeeriset arvot.

REST

Representational State Transfer on HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintoihin.

ASP.NET

ASP.NET on web-ohjelmistokehys. Sen avulla voidaan rakentaa dynaamisia web-sivuja.

Silverlight

Microsoftin kehittämä Adobe Flashin tyyppinen web-ohjelmointiympäristö.

Widget

Widget on graafisen käyttöliittymän elementti, joka näyttää tietoa tai antaa käyttäjälle tietyn tavan toimia sovelluksessa.

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. Havainnekaavio tiedonkulusta front endin ja palvelimen välillä	19
Kuvio 2. Tukikeskus yleisnäkymä -widgetit	21
Kuvio 3. Tukikeskus yleisnäkymä -kaaviot	22
Kuvio 4. Tukikeskus widget	23
Kuvio 5. Tukikeskus kaavio	24
Kuvio 6. Rajausvalikko 1/2.	27
Kuvio 7. Rajausvalikko 2/2.	27
Kuvio 8. Tukipyynnölista.	27
Kuvio 9. Esimerkki \$q-palvelun käytöstä	29
Kuvio 10. Esimerkki ng-repeat-direktiivin käytöstä	30
Kuvio 11. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 1/4.	32
Kuvio 12. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 2/4.	32
Kuvio 13. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 3/4.	33
Kuvio 14. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 4/4.	33
Kuvio 15. Uusi tukipyynnö-sivu	35
Kuvio 16. Uusi tukipyynnö Asiakastiedot-paneeli	36
Kuvio 17. Uusi tukipyynnö Sisältö-paneeli	37
Kuvio 18. Uusi tukipyynnö Perustiedot-paneeli	38
Kuvio 19. Uusi tukipyynnö Yhteyshenkilöt-paneeli	39
Kuvio 20. Uusi tukipyynnö Liite-paneeli	40
Kuvio 21. Tukipyynnö-sivu	41
Kuvio 22. Kuva tukipyynnön Liite-paneelistä jossa tukipyynnön liitekuva avattuna.	42
Kuvio 23. AngularJS:n ng-class-direktiivin käyttöä koodissa	44
Kuvio 24. Tukipyynnö Tapahtumat-paneeli	44
Taulukko 1. MVC:n osat	12
Taulukko 2. Rajausmahdollisuudet Tukipyynnö-listan eri näkymissä	31

1 JOHDANTO

Opinnäytetyönä toteutamme Lemonsoft Oy:lle uuden helpdesk-järjestelmän käyttöliittymän. Työssä käytetään useita nykyaikaisia ohjelmointitekniikoita, joista keskitymme tässä työssä pääosin AngularJS-ohjelmistokehykseen. Käymme työssä läpi sekä teoriaa käytettyjen teknologioiden ominaisuuksista, että varsinaista sivuston toteutusta. Esitämme työssä esimerkein käyttöliittymän pääkohdat sekä teknologioiden ominaisuudet, joita käytettiin niiden toteuttamiseksi tehokkaalla tavalla.

1.1 Toimeksiantaja

Toimeksiantaja on Lemonsoft Oy, joka on toiminnanohjausjärjestelmää kehittävä ohjelmistotalo. Yrityksen toimialueena on Suomi. Lemonsoftilla on noin 50 työntekijää, ja toimipisteet ovat Vaasassa, Joensuussa sekä Tampereella.

1.2 Tavoitteet

Työmme tavoitteena on tehdä toimiva käyttöliittymä helpdesk-järjestelmälle, jonka käyttö olisi mahdollisimman sujuvaa ja tehokasta. Käyttöliittymän kautta Lemonsoft Oy käsittelee ja kirjaa yritykselle saapuvat tukipyynnöt. Käyttöliittymää käyttää lähes koko yrityksen henkilöstö ja suurimmassa käytössä se on yrityksen tukiosastolla. Käyttöliittymän kautta käsiteltävä tietomäärä on melko suuri, sillä Lemonsoftille kirjataan viikoittain satoja tukipyynnöitä. Käyttöliittymästä olisi tarkoitus löytää helposti selaamalla tukipyynnöitä sekä edetä niiden käsittelyyn.

1.3 Työnjako

Työnjaon onnistuminen oli työmme toteutuksessa tärkeää pääasiassa versionhallinnan kannalta. Työstä oli löydettävä tarpeeksi erilliset sovelluksen osat, joita kukin pystyi itsenäisesti kehittämään projektin sujuvan läpiviennin sekä versionhallinnan konfliktien välttämisen varmistamiseksi. Työ jakautui siten, että Joel Moisio vastasi Tukikeskus-sivusta sekä Tukipyynnöt-listasta. Miro Salo puolestaan vastasi tukipyynnöt-sivusta sekä uuden tukipyynnön luonnin sivusta. Työn toteutusvaiheessa työnjako onnistui hyvin, ja versionhallinnan konflikteilta vältyttiin.

Työn raportointivaiheessa työnjako suoritettiin toteutuksen mukaisesti, kumpikin kirjoitti tämän työn toteutus-osiossa itse tekemistään sovelluksen osista. Näiden lisäksi työssä käytävät teoriaosuudet jaettiin siten, että Miro Salo kirjoitti AngularJS:n teoriaa käsittelevät osuudet, Joel Moisio kirjoitti Johdannon sekä muita ohjelmointiteknologioita käsittelevät osuudet. Tiivistelmä ja yhteenveto kirjoitettiin yhteistyössä.

1.4 Ohjelmointiteknologiat käyttöliittymän toteutuksessa

Työ on toteutettu osaksi Lemonsoft Oy:n onlinesovellusta, ja sen rakenne noudattaa kyseisen sovelluksen ohjelmointitapoja ja -periaatteita. Sovelluksen rakenteen arkkitehtuuri on MVC-ohjelmointiarkkitehtuurin mukainen.

Työ koostuu pääasiassa käyttöliittymän ohjelmoinnista, mutta sisältää myös muutamia palvelinpuolen elementtejä, palvelinpuolella ohjelmointikielenä on Visual Basic.

1.5 Tutkimusaiheet

Työn tarkoituksena on tutkia, kuinka saadaan luotua käytettävissä olevilla ohjelmointiteknologioilla käyttöliittymä, mikä soveltuu yrityksen tukipyynnöiden tehokkaaseen ja monipuoliseen käsittelyyn. Käyttöliittymän lisäksi työn oleellisena osana on tiedonhaku SQL Server tietokannasta aina käyttöliittymälle saakka. Tämä toteutetaan ASP.NET web-API rajapintaa käyttämällä, sekä edelleen palvelinpuolella Visual Basic -ohjelmointikielen SqlConnection nimiavaruuden toiminnoilla. ASP.NET on web-ohjelmistokehys, jonka avulla voidaan rakentaa dynaamisia web-sivuja.

Työssämme AngularJS ohjelmistokehystä käyttämällä luodut osat ohjelmoidaan TypeScript-kielellä, joka on JavaScript-kielen laajennus.

Työn tutkimusaiheet voidaan jakaa sovelluksen käytön mukaisesti kolmeen osaluueeseen. Tukikeskus-sivulla tutkimusongelmina ovat tiedon määrän rajoittaminen verkkoliikenteessä, sekä tiiviisti esitetty otsikkotason tieto tukipyynnöistä eri kriteereillä ja näiden lisäksi erinäisten mittareiden esitys aiheellisesta tukipyynnöjen

tilanteesta sekä kehityksestä. Tukikeskus-sivun päätarkoituksena on esittää tietoa, sekä mahdollistaa navigoituminen Tukipyynnö-listalle, muita käyttäjästä riippuvia toimintoja ei ole. Tukikeskus-sivun toteutuksesta kerromme lisää kappaleessa 5.2 Tukikeskus.

Sovelluksen Tukipyynnö-listalla tutkimusongelmia ovat tiedon määrän rajoittaminen verkkoliikenteessä, asynkronisten AngularJS:n \$http-palvelun kutsujen toteuttaminen synkronisesti, tietoturva tukipyynnöjen hakujen yhteydessä sekä kahden eri näkymän sisällyttäminen samaan tiedostoon. Tukikeskus-sivuun verrattaessa Tukipyynnö-lista sisältää enemmän toiminnallisuuksia, joihin käyttäjä vaikuttaa. Nämä toiminnot ovat listan suodattaminen käyttäjän valitsemilla hakukriteereillä, hakutuloksen järjestely halutun otsikkotiedon mukaisesti, sekä eteneminen yksittäisen tukipyynnön sivulle. Tukipyynnö-listan toteutuksesta kerromme lisää kappaleessa 5.3 Tukipyynnölista.

Uuden tukipyynnön luomissivulla tutkimusongelmana oli saada kaikki paneelit sopimaan yhdelle sivulle järkevään järjestykseen samantyyppisesti kuin edellisessä helpdesk-käyttöliittymässä, koska näin käyttäjien on helppo siirtyä vanhasta käyttöliittymästä uuteen eikä aikaa ei kulu niin kauan uuden opettelemiseen. Uusi tukipyynnö-sivun toteutuksesta kerromme lisää kappaleessa 5.4 Uusi tukipyynnö

Yksittäisen tukipyynnön sivulla tutkimusongelmana oli saada kaikki paneelit sopimaan yhdelle sivulle järkevään järjestykseen ja. Koska tietoa oli niin paljon, jotain tietoja täytyi saada piilotettua, että tärkeät tiedot näkyisivät sivulla paremmin. Tutkimusongelmana oli myös liitteiden esittäminen ja avaaminen sivulla ja tukipyynnön tapahtumalistan toteutus mahdollisimman selkeästi luettavaksi. Yksittäisen tukipyynnön sivun toteutuksesta kerromme lisää kappaleessa 5.5 Tukipyynnö.

1.6 Osaaminen

Projektin alussa kummankin tekijän osaaminen koostui ohjelmoinnista C#-kielellä sekä SQL-kielellä. Näiden lisäksi HTML:n ja JavaScriptin perusteet olivat hallussa.

Projektin aikana taidot kehittyivät nopeasti ja työn valmistumisen jälkeen molemmilla on sujuva osaaminen käytetyistä teknologioista, varsinkin AngularJS-ohjelmistokehyksestä.

Projektin edetessä opiskelimme itsenäisesti käytettyjä ohjelmointiteknologioita verkosta löytyvien dokumentaatioiden sekä opetusmateriaalien kautta.

2 MVC

MVC eli model-view-controller, on arkkitehtuurimalli, jonka tarkoituksena on erotella sovellus kolmeen loogiseen osaan: model, view ja controller. Jokaisen osan on tarkoitus käsitellä tiettyjä sovelluksen kehityksen näkökohtia. (Tutorialspoint 2016a)

Taulukko 1. MVC:n osat

Model	MVC arkkitehtuurin model-osa vastaa käyttäjän käsittelemän tiedon logiikasta. Tämä voi tarkoittaa tietoa, jota siirretään viewin ja controllerin välillä, tai mitä tahansa muuta bisneslogiikkaan liittyvää tietoa. (Tutorialspoint 2016 b)
View	MVC arkkitehtuurin view-osaa käytetään sovelluksen käyttöliittymän logiikkaan. (Tutorialspoint 2016 b)
Controller	MVC arkkitehtuurin controller-osat toimivat rajapintana modelin ja viewin välillä prosessoiden bisneslogiikkaa, saapuvia pyyntöjä, tiedon muokkaamista käyttäen model-osaa sekä luoden lopullisen näkymän vuorovaikutuksessa view-osien kanssa. (Tutorialspoint 2016 b)

3 ANGULARJS

AngularJS on Googlen kehittämä avoimen lähdekoodin JavaScript-ohjelmistokehys. Googlen työntekijä Miško Hevery aloitti AngularJS kehittämisen vuonna 2009, ensimmäinen versio julkaistiin kuitenkin vasta vuonna 2012.

AngularJS helpottaa yhden sivun tekniikalla toimivien sivujen toteuttamista ja säästää huomattavan määrän koodia. Tavoitteena AngularJS:llä on lisätä selaimiin tuki MVC-arkkitehtuurille. (Seshadri S, Green B 2014, 1-2)

3.1 Direktiivi

Direktiivit ovat komentoja, joilla voidaan merkitä HTML-elementti käyttäytymään tietynlaisesti. AngularJS sisältää joitain valmiita sisään rakennettuja direktiivejä kuten ng-app, ng-repeat ja ng-show. (AngularJS.org 2016a)

Tässä työssä käytämme data-ng-direktiiviä pelkän ng:n sijaan, koska se on HTML5-standardin mukaisesti validi. (Lemonsoft DevHelp2014)

3.2 Moduuli

Moduuli voidaan ajatella toimivan säiliönä, johon lisätään sovelluksen tiedot ja moduuli pitää aina sisällään sovelluksen osia kuten kontrollerit, direktiivit ja palvelut. Sovelluksen osat tulevat toimintoon vasta moduulia kutsuttaessa.

Moduulin luominen tapahtuu angular.module-komennolla. Komentoon kirjoitetaan moduulin nimi ja mahdolliset moduulin riippuvuudet eli muita moduuleita, joita kyseinen moduuli tulee käyttämään.

```
var lemonsoftApp: ng.IModule =  
    angular.module("lemonsoftApp",  
        ["ui.router", "pascalprecht.translate", "kendo.directives"]);
```

Moduuliin viitataan HTML-koodissa ng-app direktiivillä

```
<body data-ng-app="lemonsoftApp" data-ng-controller="layoutController"  
    class="appframe">
```

Kun moduuliin on viitattu yläpuolella esitetyllä tavalla, kaikki tämän moduulin koodit kuten kontrollerit ja direktiivit ovat käytettävissä tämän div-elementin sisällä. Moduuli voidaan myös lisätä vaikuttamaan laajemmalle alueelle DOM:ia lisäämälle ng-app-direktiivi esimerkiksi HTML-elementtiin. DOM on lyhenne Document Object Modelista, joka on tapa kuvata HTML:n rakennetta puuna. (Seshadri 2014, 15-16)

3.3 Kontrolleri

Kontrollerit nimensä mukaan kontrolloivat sovellusta. Kontrollerit sisältävät JavaScript funktioita, jotka esimerkiksi tuovat ja näyttävät oikean datan käyttäjälle tai ovat vuorovaikutuksessa käyttäjän toimintojen kuten napin klikkauksen kanssa.

Kontrolleri luodaan käyttämällä kontroller-funktiota, jota voidaan käyttää moduulin sisällä. Ensimmäiseen parametriin annetaan kontrollin nimi. Toiseen parametriin tulee kontrollerin määritelmä siitä, mitä se tekee ja miten.

```
lemonsoftApp.controller("helpdeskFeedbackController", helpdeskFeedbackController);
```

Kontrolleri saadaan vaikuttamaan DOM-elementissä, käyttämällä Angularin ng-controller direktiiviä siinä elementissä, johon halutaan vaikuttaa. (Seshadri 2014, 17-18)

```
<div data-ng-controller="helpdeskFeedbackController" data-ng-cloak>
```

3.4 Toistin: ng-repeat

AngularJS sisältää lukuisia sisäänrakennettuja direktiivejä, jotka tekevät AngularJS:stä erittäin tehokkaan työkalun. Yksi näistä on ng-repeat, jonka avulla listojen esittäminen ja hallinta elementin sisällä on helppoa. (Seshadri 2014, 27-29)

```
<tbody data-ng-repeat="feedback in feedbackList.feedbackList">
```

Käytännön esimerkkejä on kappaleessa 5.3 Tukipyyntölista.

3.5 Näytä/Piilota: ng-hide ja ng-show

AngularJS sisältää kaksi direktiiviä, jotka on tarkoitettu HTML-elementtien näyttämiseen ja piilottamiseen ng-show ja ng-hide. Direktiivi ng-show näyttää elementin, jos tarkasteltava muuttuja on tosi. Direktiivi ng-hide taas tekee päinvastoin, jos muuttujan arvo on tosi, direktiivi piilottaa elementin. Käytännön esimerkkejä on kappaleessa 5.3 Tukipyyntölista. (Seshadri S 2014, 27)

```
<div class="ls-panel-content" data-ng-show="feedback.showFeedbackContent">
```

3.6 Jos: ng-if

Elementti voidaan myös piilottaa ng-if-direktiivin avulla. Mikäli tarkasteltava muuttuja on tosi, direktiivi poistaa DOM-elementin kokonaan. Elementti ei edes lataudu, jos muuttujan arvo ei ole tosi. Direktiivi ng-if eroaa siis ng-hide- ja ng-show:sta niin, että hide ja show vain piilottavat elementin, joten se silti latautuu DOM:iin. Käytännön esimerkkejä kappaleessa 5.3 Tukipyyntölista. (AngularJS.org 2016b)

3.7 Luokka: ng-class

Direktiivillä voidaan dynaamisesti asettaa HTML-elementille CSS-luokka, joka vaihtuu tarkasteltavan muuttujan mukaan. Tämän helpottaa sivun ulkoasun muuttamisen tietokannasta haetun tiedon mukaan. Käytännön esimerkkejä on kappaleessa 5.5 Tukipyyntö. (AngularJS.org 2016e)

3.8 Synkroninen suoritus: \$q

AngularJS:n \$q-palvelun avulla funktioita voidaan suorittaa synkronisesti tai käyttää niiden paluuarvoja, vasta kun funktiot on suoritettu. Käytännön esimerkkejä kappaleessa on 5.3 Tukipyyntölista. (AngularJS.org 2016c)

3.9 \$http -palvelu

AngularJS:n \$http-palvelu on palvelu, jolla kommunikoidaan serverin kanssa XMLHttpRequest-olion avulla. XMLHttpRequest on API, jonka avulla siirretään

dataa sovelluksen ja serverin välillä. Sen avulla dataa voidaan hakea helposti ja päivittää vain jokin osa sivusta ilman, että koko sivua tarvitsee päivittää. (AngularJS.org 2016d)

\$http palvelu sisältää kaksi parametria ”metodi” ja ”url” joihin vaikutetaan. Metodeja joita käytetään palvelimen kyselyssä ovat:

- GET palvelimelta haetaan tieto.
- POST palvelimelle tallennetaan tieto.
- PUT palvelimella muutetaan tieto.
- DELETE palvelimelta poistetaan tieto.

4 MUUT OHJELMOINTITEKNOLOGIAT

4.1 Typescript

TypeScript on JavaScriptin laajennus, joka pitää sisällään mm. muuttujien tyyppityksen. Typescript kääntäjä kääntää sovelluksen ajossa TypeScript koodin JavaScriptiksi. (TypeScript dokumentaatio 2016 a)

4.2 Automapper

Automapper on luokkakirjasto, joka käyttää sujuvaa konfiguraatorajapintaa määrittääkseen strategian objektin yhdistämiseksi. Automapper käyttää samankaltaiseen nimeämiseen perustuvaa vastaavuusalgoritmia yhdistääkseen lähdearvot kohdearvoihin. Automapper on tarkoitettu käytettäväksi mallin projektoimisessa monimutkaisten objektimallien supistamiseksi DTO-objekteiksi (data-transfer-object), jotka soveltuvat paremmin serialisointiin, kommunikointiin sekä viestintään. Automapper tukee seuraavia alustoja:

1. .NET 4.5
2. .NET Platform Standard 1.1 & 1.3 (Automapper dokumentaatio 2016a)

4.3 ASP.NET Web API

ASP.NET Web API on kehys, jonka avulla voidaan helposti rakentaa HTTP palveluita, jotka saavuttavat laajan valikoiman selain- ja mobiilisovelluksia. ASP.NET Web API on ihanteellinen alusta REST- tyyppisten sovellusten rakentamiseen .NET ohjelmistokehyksessä. REST on lyhenne sanoista Representational State Transfer, mikä tarkoittaa HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintoihin. (ASP.NET Web API dokumentaatio 2016a)

4.4 Bootstrap

Bootstrap on nykyajan suosituin ohjelmistokehys responsiivisten web-projektien kehittämiseen. Bootstrap sisältää ominaisuuksia HTML, CSS, ja JavaScript ohjelmistokehyksiin. Se tekee front end-kehityksestä nopeampaa ja helpompaa.

Bootstrap on tarkoitettu kaiken tasoisille ohjelmoijille (Bootstrap dokumentaatio 2016 a)

4.5 Highcharts

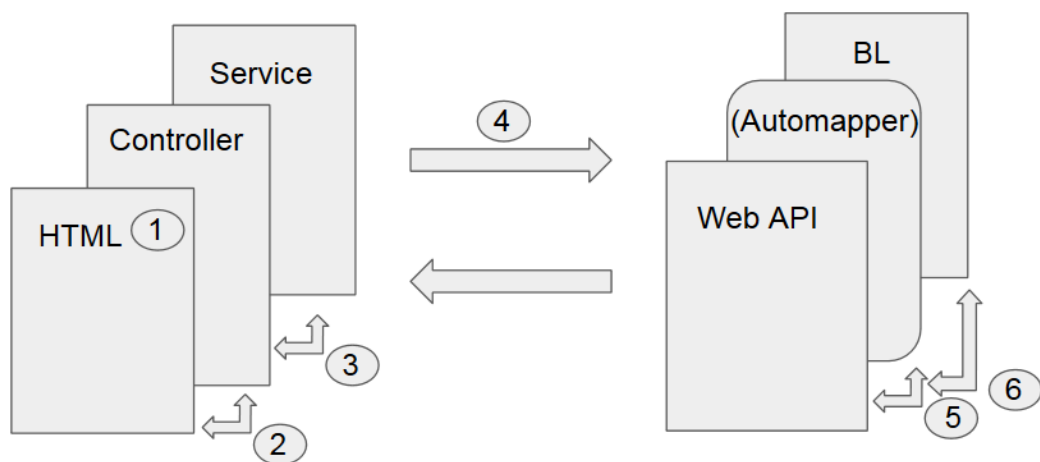
Highcharts on JavaScript kirjasto, jonka avulla voidaan lisätä web-sovellukseen interaktiivisia kaavioita. Kirjasto pitää sisällään laajan skaalan erityyylisiä kaavioita. (Highcharts dokumentaatio 2016 a)

5 TOTEUTUS

Tässä kappaleessa käymme läpi käyttöliittymän varsinaista toteutusta vaihe vaiheelta. Tarkastelemme kehityksessä ilmeneviä tutkimusongelmia, sekä niitä ohjelmointiteknologioiden osia ja ominaisuuksia, joita käytimme ongelmien ratkaisemiseksi.

5.1 Rakenne

Sovelluksen rakenne koostuu HTML-sivusta, AngularJS- kontrollerista, AngularJS-palvelusta, ASP.NET Web API:sta, Automapperista, sekä Visual Basic luokkakirjastoista. Sivun käyttöä hallinnoidaan AngularJS-kontrollerissa. Kontrollerille on asetettu riippuvuusinjektiolla kaikki ne AngularJS-palvelut, joita kyseisellä näytöllä tarvitaan. Kun halutaan suorittaa toimintoja, noudatetaan kuviossa 1 esitettyä kaavaa.



Kuvio 1. Havainnekaavio tiedonkulusta front endin ja palvelimen välillä

1. Toiminto alkaa joko ohjelmallisesti esim. sivun latautuessa tai käyttäjän toiminnasta HTML-sivulla.
2. HTML-sivulle asetettu AngularJS-kontroller käsittelee toiminnon.
 - 2.1. Mikäli toiminnon suorittaminen ei vaadi kutsuja palvelimelle, toimintoketju päättyy tähän.

3. Mikäli halutaan suorittaa kutsuja palvelimelle, AngularJS-kontroller kutsuu AngularJS-kontrollerille riippuvuusinjektiolla asetettua AngularJS-palvelua.
4. Seuraavaksi AngularJS-palvelu kutsuu ASP.NET Web API- rajapinnan metodia käyttämällä AngularJS:n http.get-metodia.
5. Jos toiminnossa muokataan jotain bisneslogiikassa esiintyvää objektia, tapahtuu Automapperin avulla tässä sovelluserroksessa muutos supistetusta DTO-objektista (data-transfer-object) rikkaampaan bisneslogiikan Visual Basic -objektiin. DTO-muutos suoritetaan aina siten, että palvelinpuolella käsittelyssä on bisneslogiikan rikkaampi objekti, ja front end-puolella käsittelyssä on tyypistetty DTO-luokka, joka usein sisältää vain propertyjä, ja joka on tarkoitettu pelkästään tiedon kuljettamiseen.
6. ASP.NET Web API:n metodista kutsutaan lopuksi varsinaista bisneslogiikan luokkaa, ja suoritetaan halutut toimenpiteet, ja lähetetään samaa reittiä pitkin front endille tarvittaessa muokatusta objektista takaisin käännetty DTO-objekti, tai vain tieto siitä, että toiminto on suoritettu.

5.2 Tukikeskus

Tukikeskus-näyttö on helpdesk-käyttöjärjestelmän keskipisteenä toimiva sivu. Näytöltä tulee nähdä widget-tyylisiltä paneeleilta tukipyynnöjä sekä kehitysideoita ennalta asetettujen hakuperusteiden mukaisesti. Widget on graafisen käyttöliittymän elementti, joka näyttää tietoa tai antaa käyttäjälle tietyn tavan toimia sovelluksessa. Widgeiteiltä tulee myös pystyä navigoitumaan tukipyynnöjen listanäytölle. Näiden widgeettien lisäksi Tukikeskus-sivulta tulee voida tarkastella yrityksen tukipyynnöjen statistiikkaa kaavioiden muodossa. Tukikeskus-sivulta voidaan myös navigoitua uuden tukipyynnön luomiseen tarkoitettulla näytölle.

Tukikeskus-sivu luotiin aikaisemmin kappaleessa 5.1 mainitun rakenteen mukaisesti. Sivun koostuu AngularJS-direktiiveinä luoduista widgeiteistä, sekä Highcharts teknologiaa soveltavina AngularJS-direktiiveinä tehdyistä kaavioista.

Direktiivien käyttö toteutetaan div-tageihin alapuolella esitetyllä tavalla.

```

<div id="widgetUndirectedFeedbackList" class="col-xs-12 col-md-6" data-ls-widget>
  <div data-ls-widget-undirected-feedback-list></div>
</div>

<div id="chart2" class="col-md-6" data-ls-editable>
  <div class="ls-panel">
    <div class="ls-panel-content">
      <div data-ls-chart-pie data-chart-number="helpdeskCentrum.currentWeekByType">
        </div>
      </div>
    </div>
  </div>
</div>

```

Kun kaikki direktiivit ovat asetettu oikein, ne muodostavat tukikeskuksen yleisnäkymän, joka esitetään kuvissa 2 ja 3. Kuvissa esiintyvistä yksittäisistä osista näytetään esimerkit myöhemmin tarkempina.

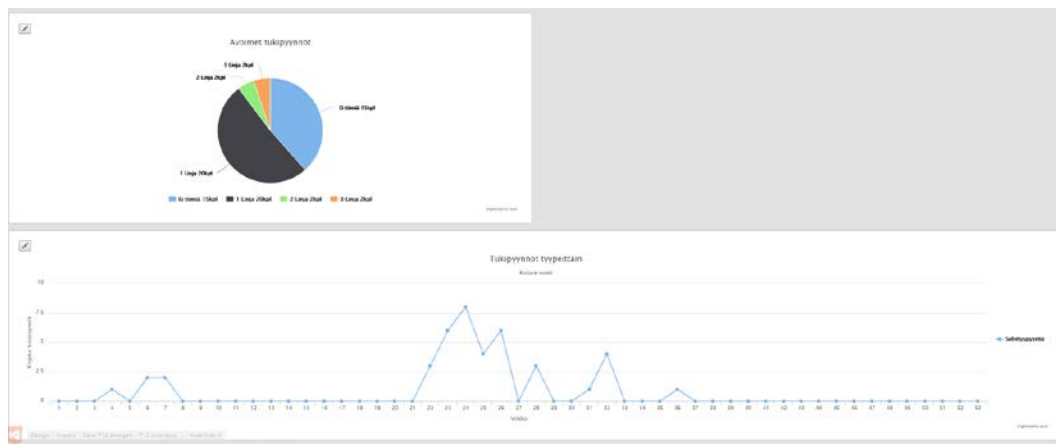
ID	Asiakas	Otsikko	Tila	Kirjasto	Päivitys
8	A Teronteknologia Oy	raportti		8.2.2016	
5	A Harjoitus	testi		30.6.2016	30.6.2016
22	B Tuttu Timo	E		26.6.2016	
44	Harjoitus	Sääntötesti joni		28.6.2016	
48	Harjoitus	Testipyyntö		13.7.2016	13.7.2016
184	Harjoitus	Tapahtumakehän selityksiä		14.7.2016	14.7.2016
201	Harjoitus	Testi		6.9.2016	

ID	Asiakas	Otsikko	Tila	Käsitellyt	Käynnä	Päivitys
6	A Teronteknologia Oy	testi		joel moisto		25.7.2016
24	A Harjoitus	huoltoinnoille ei tulosta sarjanumero		joel moisto		10.6.2016
14	B Tuttu Timo	Työkaluohjelma		Timo Arvola		16.2.2016
2	A Harjoitus	testi2		joel moisto		12.7.2016
15	B Tuttu Timo	Kangonkain tyhjillään		Timo Arvola		16.2.2016
20	B Tuttu Timo	e		joel moisto		10.6.2016
47	B Harjoitus	Käsitellyt ei tulosta sivellä		joel moisto		21.6.2016
16	C Teronteknologia Oy	a		joel moisto		19.6.2016
17	C Teronteknologia Oy	Leikkuristykäsi ei läheta sähköä tulostimelle		joel moisto		10.6.2016
18	C Teronteknologia Oy	Olikohan tämä ilmi joskus oltukin vai mitä arvelet?		joel moisto		10.6.2016

ID	Asiakas	Otsikko	Tila	Kirjasto	Päivitys
5	A Teronteknologia Oy	testi		25.7.2016	7.6.2016
24	A Harjoitus	huoltoinnoille ei tulosta sarjanumero		30.6.2016	30.6.2016
	A Harjoitus	testi2		12.7.2016	5.8.2016

ID	Asiakas	Otsikko	Tila	Käsitellyt	Käynnä	Päivitys
8	A Teronteknologia Oy	raportti				8.2.2016
15	C Teronteknologia Oy	a		joel moisto		10.6.2016
36	Harjoitus	OSANE		joel moisto		14.6.2016

Kuvio 2. Tukikeskus yleisnäkymä -widgetit






Kuvio 3. Tukikeskus yleisnäkymä -kaaviot

Jotta tiedon määrä verkkoliikenteessä ei kasvaisi liian suureksi esim. mobiiliverkkoa käyttämällä, haetaan kullekin widgetille rajattu määrä tukipyyntöjä.

Eri päätelaitteiden sujuva käyttö taataan Bootstrap-teknologiaa käyttämällä. Tyylit on muokattu yritykselle sopiviksi yrityksen tyyleistä vastaavien työntekijöiden toimesta.

Widgeteissä ja Highcharts-kaavioissa käytämme Lemonsoftin arkkitehtuuritiimin luomia kantatoteutuksia. Widgeteilta voidaan tarkastella tukipyyntöjä ennalta asetettujen hakuperusteiden mukaisesti.

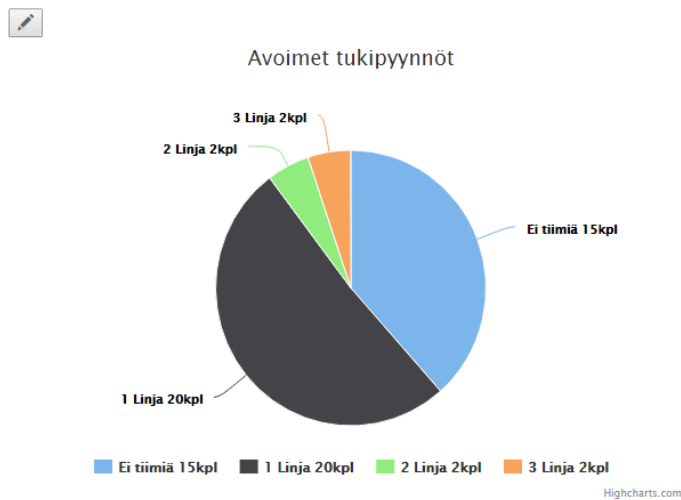
Kohdistamattomat tukipyynnöt 

ID		Asiakas	Otsikko	Tila	Kirjattu	Päivitetty
8		A Terontekniikka Oy	Raportti		8.2.2016	
5	 	A Harjoitus	testi		30.6.2016	30.6.2016
22	 	B Tuttu Timo	g		2.6.2016	
44		Harjoitus	Sisäinen testi JOEL		28.6.2016	
48		Harjoitus	Testipyyntö		13.7.2016	13.7.2016
144		Harjoitus	Tapahtumaketjun selvitystä		14.7.2016	14.7.2016
281		Harjoitus	Test		6.9.2016	

Kuvio 4. Tukikeskus widget

Widgetien kantatoteutuksen lisäksi widgetin koodissa hallinnoidaan tukipyyntöjen haut palvelimelta sekä asetetaan Tukipyyntö-lista, jolle pystytään navigoitumaan. Tiedonhaun toimintojen lisäksi tukikeskus-widgeteille on asetettu muuttuja kertoamaan, miltä widgetiltä saavutaan, kun Tukipyyntö-listalle edetään. Tämän muuttujan avulla voidaan Tukipyyntö-listalla asettaa oletus hakuperusteet vastaamaan widgetillä esitettyä rajattua listaa.

Highcharts kaavioista nähdään yrityksen tukipyyntöjen ajantasainen tilanne sekä lähimenneisyyden kehitys.



Kuvio 5. Tukikeskus kaavio

Widgeteille sekä kaavioille olennainen osa on tiedon sisällön hakeminen SQL Server- tietokannasta.

Widgettien tiedonhakua varten luotiin SQL Viewit alapuolella esitetyllä tavalla.

```
CREATE VIEW dbo.v_open_feedbacklist
AS
SELECT feedback.*,
persons.person_name
AS 'person_name',
codelist_abc_groups.codelist_id
AS 'codelist_abc_group_id',
codelist_abc_groups.codelist_description
AS 'codelist_abc_group_description',
(SELECT COUNT(*) FROM feedback_appendixs
WHERE feedback_appendixs.feedback_id = feedback.feedback_id)
AS 'appendix_count'
FROM feedback
```



```

LEFT JOIN persons ON
feedback.person_id = persons.person_id
LEFT JOIN customers on
  feedback.feedback_customer_number = customers.customer_number
LEFT JOIN customer_dimensions on
  customers.customer_id = customer_dimensions.customer_id
LEFT JOIN codelist_abc_groups on
| codelist_abc_groups.codelist_id = customer_dimensions.customer_abc
WHERE feedback.feedback_id > 0
AND feedback.feedback_type IN (2,3)
AND feedback.feedback_state NOT IN (-1,15,90)
GO

```

Yläpuolella esitetty SQL-view hakee kaikki avoimet tukipyynnöt. Tulosjoukon ko-
koa rajataan bisneslogiikan koodissa, jossa viewiä käytetään.

Highcharts kaavioiden tiedonhakua varten luotiin talletetut SQL-Proseduurit ala-
puolella esitetyllä tavalla.

```

CREATE PROCEDURE [dbo].[up_sel_open_feedbacks_by_team]
  @iPersonNumber AS INT
AS
|-- Haetaan käyttäjän kielikoodi
-- Kielikoodin mukaan käännetään tukipyynnön tyyppinimi
DECLARE @strLang AS NVARCHAR(10) =
ISNULL((SELECT language_code FROM persons
WHERE person_number = @iPersonNumber), 'FIN')
SELECT
CASE WHEN fb.feedback_team = 0 OR fb.feedback_team IS NULL
THEN dbo.translate('Ei tiimiä', @strLang) + ' ' +
CONVERT(VARCHAR(10),COUNT(*)) + dbo.Translate('kpl', @strLang)
ELSE dbo.translate(teams.team_description, @strLang) + ' ' +
CONVERT(VARCHAR(10),COUNT(*)) + dbo.Translate('kpl', @strLang) END, COUNT(*)
FROM feedback fb
LEFT JOIN teams ON fb.feedback_team = teams.team_id
WHERE fb.feedback_state NOT IN (15,90)
AND fb.feedback_type IN (2,3)
GROUP BY teams.team_description, fb.feedback_team
GO

```

Yläpuolella esitetty talletettu SQL-proseduuri hakee avointen tukipyyntöjen luku-
määrän ja ryhmittelee ne tiimeittäin. Proseduuria käytetään tukikeskuksen kaavi-
ossa, joka esittää avointen tukipyyntöjen jakauman tiimeittäin.

5.3 Tukipyyntölista

Sovelluksen Tukipyyntö-listalla käyttäjä hakee itse valitsemiensa hakuperusteiden mukaisesti listan tukipyynnöistä. Tukipyyntö-listan näkymä on hieman erilainen riippuen siitä, ollaanko hakemassa tukipyyntöjä tai kehitysideoita.

Kummankin näkymän käyttöperiaate sekä niissä käsiteltävät objektit ovat samoja, joten käytämme Tukipyyntö-listan näkymissä samaa HTML-tiedostoa. Tukipyyntö-listan näkymä määritellään AngularJS:n ng-if-direktiiviä käyttämällä. Valitsimme ng-if-direktiivin ng-show/ng-hide direktiivien sijasta, koska näytettävä sisältö ei muutu yhden käynnin aikana, joten turhat elementit DOM:ssa voidaan jättää luomatta. Direktiivin ng-if käytöstä esimerkki alapuolella.

```
<div class="col-xs-12" data-ng-if="feedbackList.isFeedbackView">
```

Tukipyyntö-lista koostuu koko näytön levyisestä listanäkymästä, jolla esitetään tukipyynnön perustietoja sekä piilotettavasta valikosta, josta käyttäjä pystyy muuttamaan hakukriteereitä. Rajausvalikon esittämisessä ja piilottamisessa olemme käyttäneet Bootstrap JavaScript-kirjaston collapse-direktiiviä.

Kuvissa 6,7 ja 8 nähdään tukipyyntölistan näkymästä ensin piilotettavissa oleva rajausvalikko, ja sen jälkeen itse tukipyyntölista.

Tyypisuodatus

Selvityspyyntö Virhe Raportti

Tilasuodatus

Kirjattu Hyväksytty Työjonossa Hylätty Aloitettu Odottaa Valmis

Päivämääräsuodatus

Alkaen

Päättyen

Kuvio 6. Rajausvalikko 1/2.

Tilatarkennesuodatus

Odottaa asiakasta Odottaa sisäistä

Käsittelijäsuodatus

James Bond
Koodaaja Kalle
Miro Salo
webb

Tiimisuodatus

Vaasa Tiimi
Tampere
Helsinki
Seinäjoki

Ilman käsittelijää

Muut suodatukset

Tänään kirjatut Faq Muuttuneet Lukitut

Kuvio 7. Rajausvalikko 2/2.

Tukipyynnötilista Suodata 1/1

ID IT	Asiakas IT	Otsikko IT	Tila IT	Käsittelijä IT	Kirjattu IT	Päivitetty IT
+ 6	A Terontekniikka Oy	testi	●	Joel Moisio	25.1.2016	7.6.2016
+ 24	A Harjoitus	huoltoilmoitukselle ei tulostu sarjanumero	○	Joel Moisio	30.6.2016	30.6.2016
+ 2	A Harjoitus	testi2	○	Joel Moisio	12.7.2016	5.8.2016
+ 42	B Harjoitus	Kassakuitit ei tulostu oikein	○	Joel Moisio	21.6.2016	
+ 20	B Tuttu Timo	e	○	Joel Moisio	10.6.2016	2.6.2016
+ 46	C Teronteknologia Oy	Tänne ei vielä sisäisiä kirjatakaan	○	Joel Moisio	28.6.2016	

Kuvio 8. Tukipyynnötilista.

Listan tulosjoukosta pystytään tarkastelemaan koko ajan esillä olevia tukipyynnöiden perustietoja. Tukipyynnötilistan sivutustyökalu on sivun oikealla ylä laidassa, tästä voidaan selailla koko tulosjoukon eri sivujen tukipyynnöitä. Suodatin-nappi näyttää/piilottaa rajausvalikon, ja päivitys nappi päivittää listan. Suodata-kenttään kirjoitettu hakuteksti suorittaa ns. pikahaun, joka ohittaa muut rajausvalinnat ja hakee pelkästään hakusanaa vastaavia tukipyynnöitä. Tulosjoukon riviä klikkaamalla nähdään lisätietoa yhdestä tukipyynnöstä ja listan vasemmalla laidalla esiintyvää +-

näppäintä klikkaamalla nähdään tukipyynnön tapahtumaketjun 10 viimeisintä tapahtumaa. Tukipyynnön ID-tieto toimii linkkinä Tukipyyntö-sivulle. Piilotettavien tietojen esittämiseen olemme käyttäneet AngularJS:n ng-show direktiiviä. Direktiivin ng-show käytöstä esimerkki alapuolella.

```
<tr data-ng-show="feedback.Show_text">
```

Tukipyyntö-listalle saavuttaessa haetaan ensin palvelimelta hakutuloksen rajaamisen vaihtoehdot, jonka jälkeen suoritetaan ensimmäisen tulosjoukon haku oletus hakuperusteiden mukaisesti. AngularJS:n \$http-palvelukutsut ovat asynkronisia, joten tavallisesti sovellus etenisi front end-puolella AngularJS:n kontrollerissa seuraavaan metodiin, vaikka edelliseen ei ole vielä saatu vastausta palvelimelta. Tämä aiheuttaisi ongelman, sillä sovellus yrittäisi hakea ensimmäistä tukipyyntöjen tulosjoukkoa tyhjillä hakukriteereillä, koska rajausvaihtoehtoja ei ole vielä saatu palvelimelta. Tämän ongelman ratkaisemiseksi käytämme AngularJS:n \$q-palvelua, jonka avulla voidaan ensin suorittaa \$http-palvelukutsuja ja odottaa niiden valmistumista, jonka jälkeen siirrytään vasta ohjelmakoodissa eteenpäin. AngularJS:n \$q-palvelun käytöstä esimerkki kuviossa 9.

```

this.$q.all([
  this.ProductManagementService.getIntranetOnlineTypeList().success((data) => {
    this.typeList = <Array<FeedbackTypeExtensionModel>>data;
  }),
  this.ProductManagementService.getFeedbackStateList().success((data) => {
    this.stateList = <Array<FeedbackStateExtensionModel>>data;
  }),
  this.HumanResourcesService.GetFeedbackHandlerList().success((data) => {
    this.handlerList = data;
  }),
  this.HumanResourcesService.GetTeamList().success((data) => {
    this.teamList = data;
  })
]).then(() => {
  if (this.$stateParams.widgetOrigin) {
    this.setDefaultFilters(this.$stateParams.widgetOrigin);
  } else {
    if (this.isFeedbackView) {
      this.getFeedbackList(false);
    } else {
      this.getDevelopmentIdealList(false);
    }
  }
});

```

Kuvio 9. Esimerkki \$q-palvelun käytöstä

Yläpuolella esiintyvässä kuviossa haetaan ensin rajausvaihtoehdot, ja odotetaan, että metodit valmistuvat. Tämän jälkeen edetään tukipyyntöjen tulosjoukon hakemiseen.

Rajausvalikon muuttujat ovat sivun AngularJS-kontrollerissa listoina sekä boolean tyyppisinä arvoina. Niiden esittämiseksi ja hallitsemiseksi tehokkaalla tavalla HTML-sivun puolella käytämme AngularJS:n ng-repeat-direktiiviä. Tämä esitetään alapuolella olevassa kuviossa.

```
<div class="row">
|   <div class="col-lg-12">
|     <h1>{{'HD.FeedbackList.StateFiltering' | translate}}</h1>
|     <ul class="list-inline">
|       <li data-ng-repeat="state in feedbackList.stateList">
|         <div class="form-group">
|           <div class="checkbox">
|             <label>
|               <input type="checkbox" data-ng-model="state.isChecked" />
|                 {{state.Feedback_state_name}}
|             </label>
|           </div>
|         </div>
|       </li>
|     </ul>
|   </div>
</div>
```

Kuvio 10. Esimerkki ng-repeat-direktiivin käytöstä

Kuviossa 10 käydään läpi lista, joka sisältää tukipyynnön tilan rajausvaihtoehdot ja luodaan niille valintalaatikot. Luodun valintalaatikon tekstiksi asetetaan tilan nimi, ja valinnan arvo tietosidotaan tila-muuttujan isChecked-booleen arvoon.

Käytettävät rajausmahdollisuudet vaihtelevat riippuen sivun näkymästä. Mikäli ollaan hakemassa tukipyyntölistaa, tulosjoukon rajaamisen vaihtoehdot ovat laajemmat kuin jos ollaan hakemassa kehitysidealistaa.

Taulukko 2. Rajausmahdollisuudet Tukipyyntö-listan eri näkymissä

Rajaus	Tukipyyntölista	Kehitys-idealista
Tyyppi	X	-
Tila	X	X
Tilatarkenne	X	-
Päivämäärä	X	-
Tiimi	X	-
Käsittelijä	X	X
Tänään kirjattu	X	-
FAQ	X	-
Muuttunut	X	-
Lukittu	X	-

Palvelimelle suoritettavissa hauissa tiedon määrän rajoittaminen on toteutettu hakemalla listat sivutettuina. SQL-kysely on rakennettu siten, että annetuista hakukriteereistä haetusta tulosjoukosta valitaan vain tietyn kokoinen osa tietystä kohdasta. Tämä osa ja kohta, mistä se otetaan, riippuvat käyttäjän sen hetkisestä sivun numerosta sekä sivun koosta.

```

Dim oSB As New System.Text.StringBuilder
oSB.Append("SELECT * FROM (SELECT ROW_NUMBER() OVER
(ORDER BY v_feedbacklist.feedback_error_type DESC,
CASE WHEN codelist_abc_groups.codelist_id IS NULL THEN 2 ELSE 1 END,
codelist_abc_groups.codelist_id, ")
oSB.Append(iSortField.ToString())
oSB.Append(" ")
oSB.Append(iSortDirection.ToString())
oSB.Append(" ) AS rownum, v_feedbacklist.*, codelist_abc_groups.codelist_description
AS 'codelist_abc_group_description', ")
oSB.Append("(SELECT COUNT(*) FROM feedback_appendixs
WHERE feedback_appendixs.feedback_id = v_feedbacklist.feedback_id)
AS 'appendix_count' From v_feedbacklist ")
oSB.Append("LEFT JOIN customers On v_feedbacklist.feedback_customer_number =
customers.customer_number ")
oSB.Append("LEFT JOIN customer_dimensions ON customers.customer_id =
customer_dimensions.customer_id ")
oSB.Append("LEFT JOIN codelist_abc_groups ON codelist_abc_groups.codelist_id =
customer_dimensions.customer_abc ")
oSB.Append("WHERE v_feedbacklist.feedback_id > 0 ")
oSB.Append("AND v_feedbacklist.feedback_type != 1 ")
oSB.Append("AND v_feedbacklist.feedback_state != -1")

```

Kuvio 11. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 1/4.

```

If colTypes IsNot Nothing AndAlso colTypes.Count > 0 Then
    strTypeFilter = String.Join(", ", colTypes)
    oSB.Append(" AND v_feedbacklist.feedback_type IN (Select innumber
FROM dbo.f_commalist_to_tbl(@strTypeFilter)) ")
End If
If colStates IsNot Nothing AndAlso colStates.Count > 0 Then
    strStateFilter = String.Join(", ", colStates)
    oSB.Append("AND v_feedbacklist.feedback_state IN (Select innumber
FROM dbo.f_commalist_to_tbl(@strStateFilter)) ")
End If
If colHandlers IsNot Nothing AndAlso colHandlers.Count > 0 Then
    strHandlerFilter = String.Join(", ", colHandlers)
    oSB.Append("AND ")
    If (bWithoutHandler) Then
        oSB.Append("( ")
    End If
    oSB.Append("person_id IN (Select innumber
FROM dbo.f_commalist_to_tbl(@strHandlerFilter)) ")
    If (bWithoutHandler) Then
        oSB.Append("OR (person_id IS NULL OR person_id = 0)) ")
    End If
End If
End If

```

Kuvio 12. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 2/4.


```

If colHandlers Is Nothing Or colHandlers.Count = 0 AndAlso (bWithoutHandler) Then
    oSB.Append("AND (person_id IS NULL OR person_id = 0) ")
End If
If colTeams IsNot Nothing AndAlso colTeams.Count > 0 Then
    strTeamFilter = String.Join(", ", colTeams)
    oSB.Append("AND v_feedbacklist.feedback_team IN (Select innumber
FROM dbo.f_commalist_to_tbl(@strTeamFilter)) ")
End If
If Not String.IsNullOrEmpty(strSearchWord) AndAlso IsNumeric(strSearchWord) Then
    oSB.Append("AND v_feedbacklist.feedback_id = @strSearchWordFilter ")
ElseIf Not String.IsNullOrEmpty(strSearchWord) Then
    strSearchWord = "%" & strSearchWord & "%"
    oSB.Append("AND (v_feedbacklist.feedback_caption
LIKE @strSearchWordFilter OR v_feedbacklist.feedback_text
LIKE @strSearchWordFilter OR v_feedbacklist.feedback_customer_name
LIKE @strSearchWordFilter) ")
End If
oSB.Append("AND (@dtDateFilterFrom IS NULL
OR v_feedbacklist.feedback_date >= @dtDateFilterFrom) ")
oSB.Append("AND (@dtDateFilterTo IS NULL
OR v_feedbacklist.feedback_date <= @dtDateFilterTo) ")

```

Kuvio 13. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 3/4.

```

If bWaitingCustomer Or bWaitingInternal Then
    If bWaitingCustomer AndAlso bWaitingInternal Then
        oSB.Append("AND v_feedbacklist.feedback_state_specifier IN (10,20) ")
    ElseIf bWaitingCustomer Then
        oSB.Append("AND v_feedbacklist.feedback_state_specifier = 10 ")
    Else
        oSB.Append("AND v_feedbacklist.feedback_state_specifier = 20 ")
    End If
End If
If bOnlyToday Or bFaq Or bChanged Or bLocked Then
    Dim colConditions As New List(Of String)
    If bOnlyToday Then
        colConditions.Add("v_feedbacklist.feedback_date >= @dtToday")
    End If
    If bFaq Then
        colConditions.Add("v_feedbacklist.feedback_faq_bit = 1")
    End If
    If bChanged Then
        colConditions.Add("v_feedbacklist.feedback_changed_bit = 1")
    End If
    If bLocked Then
        colConditions.Add("v_feedbacklist.feedback_locked_bit = 1")
    End If
    oSB.Append("AND (")
    oSB.Append(String.Join(" OR ", colConditions))
    oSB.Append(") ")
End If
oSB.Append(") AS subselect WHERE rownum BETWEEN @iPageStart AND @iPageEnd")

```

Kuvio 14. Sivutetun SQL-kyselyn luonti bisneslogiikan koodissa 4/4.

Kuvioissa 11,12,13 ja 14 esitetään listahaun SQL-kyselyn luontia varten tehty koodi. Koodissa käydään läpi kaikki rajausvalinnat ja niiden perusteella kootaan StringBuilder objektilla SQL-kysely, jolla haetaan koko tulosjoukosta halutun sivun mukainen joukko, jonka koko määrittyy sivun koon mukaisesti. Sivutus toteutetaan hakemalla tietokantataulun tietojen lisäksi rivin numero yhdelle tietueelle, ja sivun numeron sekä sivun koon mukaan voidaan valita tulosjoukkoon mukaan vain tietyt rivit. Kuviosarjan ensimmäisessä osassa näkyy myös ENUM:n käyttö tulosjoukon järjestelyssä. ENUM on lyhenne Enumerated type:stä ja sen avulla voidaan määrittellä teksti arvolle numeeriset arvot.

Tietoturva hakutoiminnoille taataan liittämällä käyttäjän vapaa hakusana SQL-komennon muuttujalistaan eikä suoraan itse kyselyyn. Tämän lisäksi otsikkotason klikkauksesta aiheutuva tulosjoukon järjestely toteutetaan ENUM:a käyttämällä. Otsikot ovat merkkijonoja ja niille on asetettu numeeriset arvot vastaamaan kyseistä merkkijonoa. Muut hakukriteerit ovat avain-arvo-objekteja, joten ne liitetään osaksi SQL-hakua myös numeerisina, eikä niistä näin ollen muodostu SQL-injektioriskiä.

5.4 Uusi tukipyyntö

Uusi tukipyyntö-sivulla helpdesk-järjestelmään luodaan uusia tukipyyntöjä.

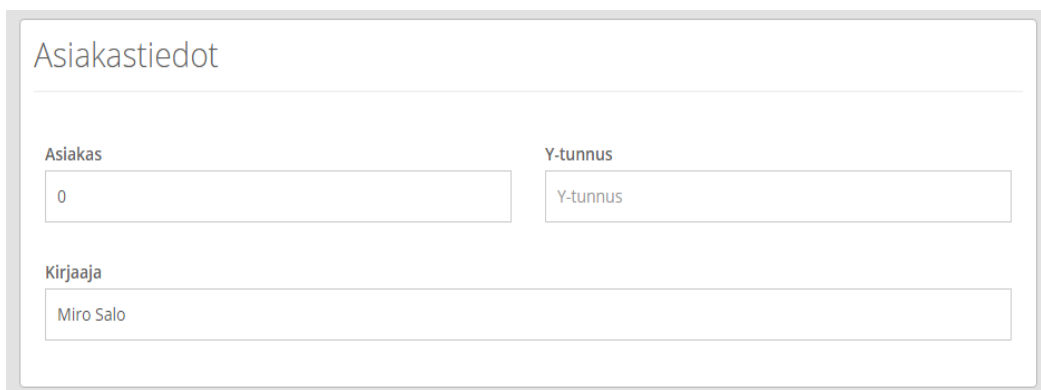
Uuden tukipyynnön luominen pitää onnistua mahdollisimman nopeaa ja helposti, joten näkymä on suunniteltu niin, että henkilön on helppo lisätä tai muuttaa asiakkaasta tietoja sitä mukaan, kun asiakas niitä kertoo. Näkymään on vaikuttanut hie-
man vanhan helpdesk-käyttöliittymän ulkoasu ja palkit on yritetty lajitella siinä jär-
jestyksessä, missä asiakas niitä yleensä kertoo. Tästä syystä näyttö ja kaikki teksti-
laatikot on luotu yhdelle sivulle, joten järjestelmää käyttävällä henkilöllä ei kulu
aikaa oikean sivun etsimiseen ja vaihteluun. Uusi tukipyyntö-sivun lopputulema
kuviossa 15.

Kuvio 15. Uusi tukipyynnö-sivu

Uuden tukipyynnön vaadittavat tiedot ovat otsikko, sisältö ja uuden tukipyynnön on sisällettävä ainakin yksi yhteyshenkilö. Oikeassa alareunassa sijaitseva tallennanappi ilmestyy näkyviin vasta sen jälkeen, kun vaadittavat tiedot on täytetty ja lisätty.

Uuden tukipyynnön luonnissa aloitetaan Asiakastiedot-paneelilla, joka on sijoitettu vasempaan yläkulmaan, koska tästä yleensä uutta tukipyynnöä aletaan täyttämään. Asiakas siis soittaa ja kertoo, mistä yrityksestä on.

Ensimmäinen tekstilaatikko asiakas on todellisuudessa hakukenttä. Kenttään kirjoitetaan esimerkiksi ”Lemon” ja haku tulos antaa tulokseksi kaikki tietokantaan kirjatut yritykset, joiden nimet alkavat Lemon. Y-tunnus kenttään ei henkilö voi itse kirjoittaa mitään, sen järjestelmä hakee itse tietokannasta, kun asiakaskenttään on valittu asiakas. Kirjaaja kenttään järjestelmä hakee automaattisesti kirjautuneen käyttäjän nimen.



Asiakastiedot

Asiakas: 0

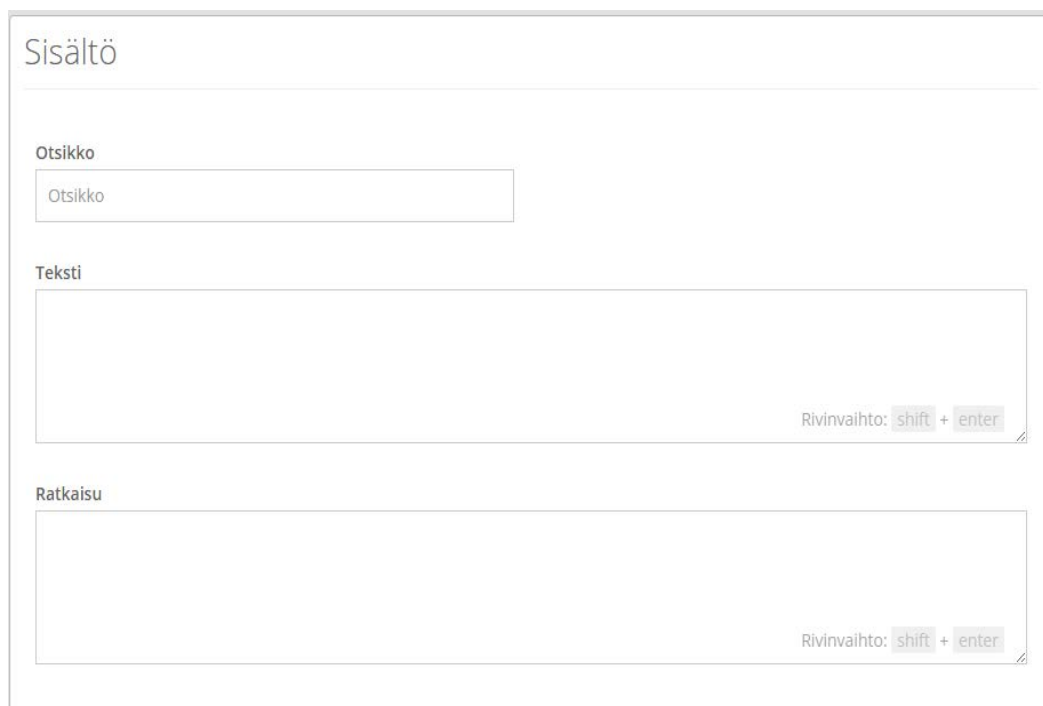
Y-tunnus: Y-tunnus

Kirjaaja: Miro Salo

Kuvio 16. Uusi tukipyyntö Asiakastiedot-paneeli

Asiakastiedot-paneelin alapuolella on uuden tukipyynnön Sisältö-paneeli. Sisältö-paneeliin kirjataan tukipyynnön sisältöä eli mikä on asiakkaan varsinainen ongelma tai kysymys, mihin apua tarvitaan.

Otsikko tekstilaatikkoon lisätään tukipyynnölle jokin ongelmaa kuvaava nimi ja tällä tukipyyntö esimerkiksi löydetään helposti tukikeskuksen lista widgeiteistä. Teksti tekstikenttään tulee itse ongelma tarkemmin, joten teksti voi olla pitkäkin, tästä syystä on myös suurempi kokoinen. Ratkaisu tekstikenttään tulee tukipyynnön sisältämään ongelmaan tai kysymykseen vastaus. Ratkaisu-tekstikenttä esitetään jo uusi tukipyyntö -näytössä, koska ongelmaan voi löytyä vastaus jo näyttöä täyttäessä. Ratkaisu-kentän voi kuitenkin jättää halutessa myös tyhjäksi.



Sisältö

Otsikko

Otsikko

Teksti

Rivinvaihto: shift + enter

Ratkaisu

Rivinvaihto: shift + enter

Kuvio 17. Uusi tukipyyntö Sisältö-paneeli

Oikeassa reunassa on uuden tukipyynnön Perustiedot-paneeli. Perustiedot-paneeliin käyttäjä lisää tukipyynnön tietoja ja näiden tietojen avulla järjestelmä kategorisoi tukipyynnön oikeaan tukikeskuksen widgettiin. Näin tukipyyntö näkyy helpommin oikealle henkilölle, joka osaa tukipyynnön mahdollisesti ratkaista.

Tyyppi ja lähde hakukenttään järjestelmä hakee automaattisesti suosituimman vaihtoehdon valmiiksi. Käsittelijä-hakukenttään järjestelmä hakee automaattisesti järjestelmään kirjautuneen käyttäjän. Tiimi-hakukenttään järjestelmä hakee käsittelijän tiimin. Vasteaika haetaan automaattisesti tietokannasta tukipyyntöön lisätyn asiakkaan mukaan. Perustiedot-paneelissa kaikki laatikot paitsi vasteaika ovat hakukenttiä, joka hakee vaihtoehdot tietokannasta sitä mukaan, kuin tekstikenttään kirjoitetaan. Viimeisenä Perustiedot-paneelissa on valintaruutu, jonka ruksaaminen lukitsee tukipyynnön tallennuksen jälkeen, jolloin ei tukipyynnön muokkaaminen ole mahdollista ennen kuin ruksi on otettu pois.

Perustiedot

Tyyppi <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Selvityspyyntö"/>	Virheluokka <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>
Versio <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>	Kategoria <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>
Tuote <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>	Projekti <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>
Käsittelijä <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Miro Salo"/>	Tiimi <input style="width: 90%; border: 1px solid #ccc;" type="text" value="3 linja"/>
Prioriteetti <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Hae"/>	Lähde <input style="width: 90%; border: 1px solid #ccc;" type="text" value="Puhelu"/>
Vasteaika <input style="width: 90%; border: 1px solid #ccc;" type="text" value="0"/>	
<input type="checkbox"/> Lukittu	

Kuvio 18. Uusi tukipyyntö Perustiedot-paneeli

Vasemmalla alareunassa on uuden tukipyynnön Yhteyshenkilöt-paneeli, johon tulevat tukipyynnön asiakkaan yhteyshenkilöiden tiedot: nimi, sähköpostiosoite, puhelinnumero.

Yhteyshenkilöt-paneelin yläosassa on tukipyyntöön lisätyt yhteyshenkilöt ja heidän tiedot listattuna. Listassa oikealla on x-nappi, joka poistaa yhteyshenkilön. Yhteyshenkilöt-paneelin alaosassa on tekstikentät joihin uuden yhteyshenkilön tiedot lisätään. Vaihtoehtona on myös hakukenttä, josta voi hakea asiakkaan yhteyshenkilöitä tietokannasta, joka lisää automaattisesti yhteyshenkilön tiedot kenttiin. Mikään tämän paneelin tekstikenttä ei ole pakollinen tieto, joten kaikkia tietoja yhteyshenkilöstä ei tarvitse tietää. Lisää-nappi lisää uuden yhteyshenkilön listalle. Lisää minut yhteyshenkilöksi-nappi lisää kirjautuneen käyttäjän automaattisesti yhteyshenkilöksi listalle.

Yhteyshenkilöt

Nimi	Sähköpostiosoite	Puhelinnumero
Matti Mallikas	✉ Matti.Mallikas@gmail.com	☎ 040 1234 567

Haku

Hae

Nimi

Nimi

Sähköpostiosoite

Sähköpostiosoite

Puhelinnumero

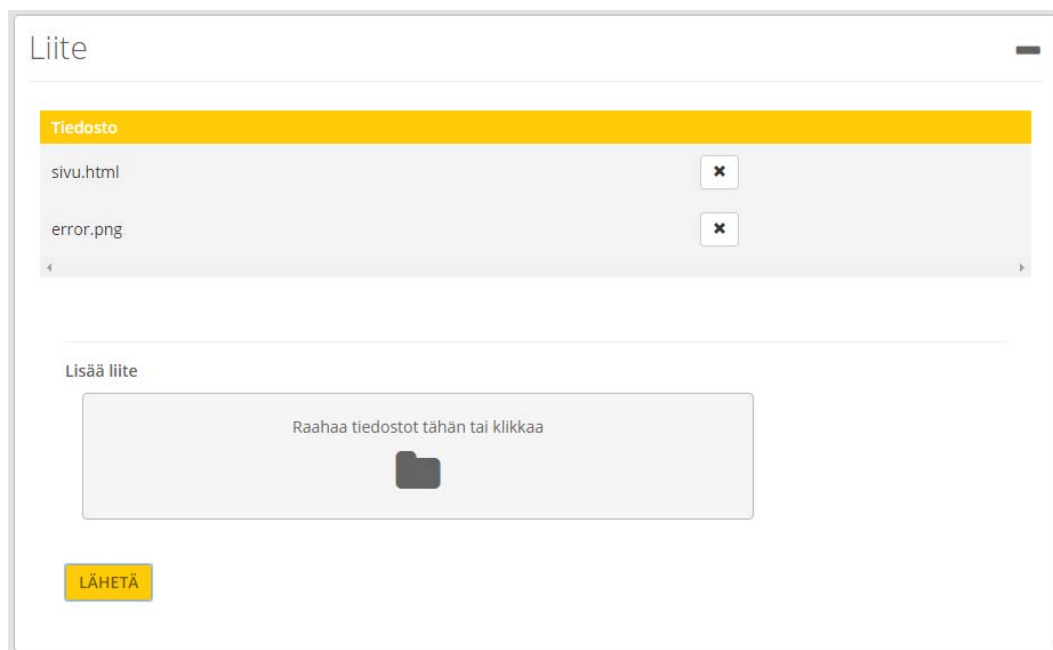
Puhelinnumero

LISÄÄ LISÄÄ MINUT YHTEYSHENKILÖKSI

Kuvio 19. Uusi tukipyyntö Yhteyshenkilöt-paneeli

Oikeassa alareunassa on uuden tukipyynnön Liite-paneeli. Liite-paneeliin lisätään tukipyynnön liitteet.

Liite-paneelin yläosassa on lisätyt liitteet listana, joissa on vasemmalla tiedoston nimi ja tiedostomuoto ja oikealla x-nappi, jolla liite poistetaan listalta. Liite-paneelin alaosassa on uuden liitteen lisäys, joka toimii ”raahaa ja pudota”-menetelmällä eli tiedosto raahataan laatikon päälle, jolloin lisättävät liitteet näkyvät listana. Liitteitä on mahdollista lisätä monta kerralla, mutta yhden liitteen kokoraja on kaksi megatavua. Lähetä-nappia painamalla tiedostot lisätään liitelistaan.



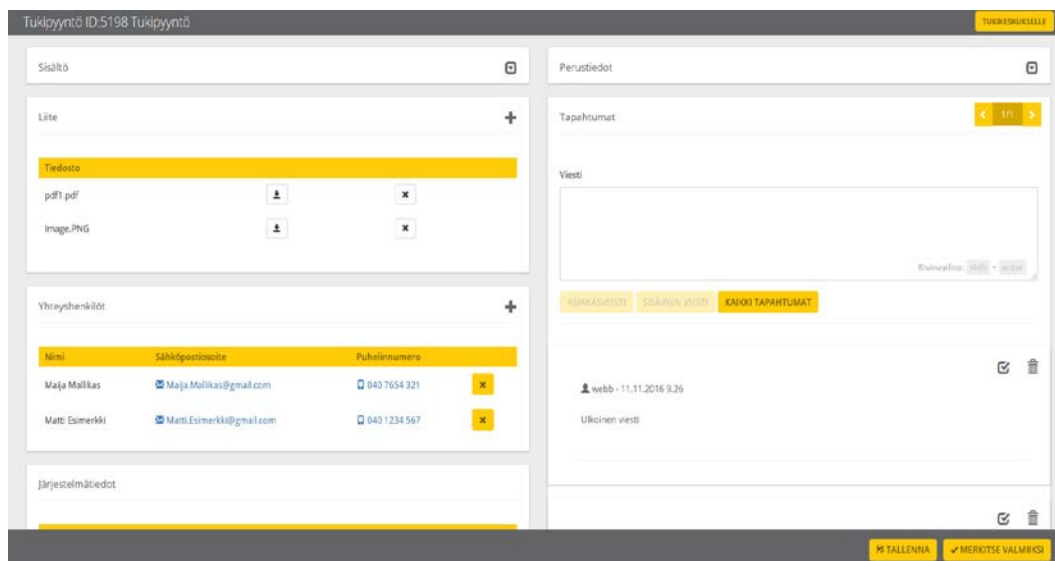
Kuvio 20. Uusi tukipyyntö Liite-paneeli

5.5 Tukipyyntö

Tukipyyntö-sivu on olemassa olevan yksittäisen tukipyynnön käsittelyyn ja muokkaamiseen käytettävä sivu.

Yksittäistä tukipyyntöä täytyy päästä käsittelemään ja kysyä asiakkaalta mahdollisesti lisätietoja viesteillä, tämä sivu on suunniteltu niin, että se olisi mahdollisimman helppoa ja sujuvaa. Tukipyyntö saattaa myös käydä monella eri henkilöllä yrityksen sisällä, joten tukipyynnössä näkyy myös tukipyynnön historia tapahtumina.

Yksittäisen tukipyynnön sivun ulkoasu on hyvin samanlainen kuin uusi tukipyyntö-sivu, jotta molempia olisi yhtä helppo käyttää. Tämä sivu sisältää kuitenkin enemmän sisältöä, joten joitain muutoksia on täytynyt tehdä ja kuten uusi tukipyyntö-sivu, niin on tämäkin sivu yritetty luoda mahdollisimman samankaltaiseksi kuin edellinen helpdesk-käyttöliittymä.



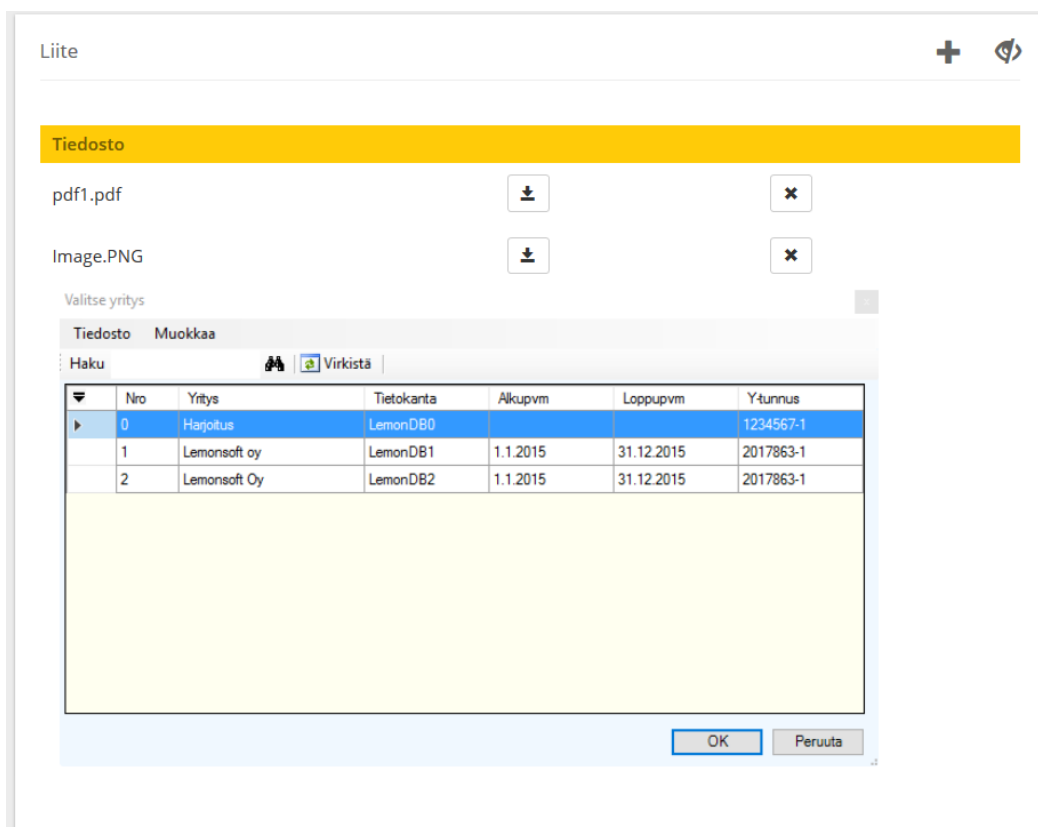
Kuvio 21. Tukipyyntö-sivu

Tukipyyntö-sivu sisältää paljon tietoa ja jotkut tiedot ovat tärkeämpiä kuin toiset, joten tilaa piti jotenkin saada tiivistettyä ja tukipyynnön käsittelyssä tärkeimmille tiedoille saada enemmän tilaa. Tähän ratkaisuna käytettiin AngularJS:n ng-show- ja ng-hide-direktiiviä. Tietoja joiden ei käsittelyn aikana tarvitse aina olla näkyvissä, kuten perustiedot ja sisältö, voidaan piilottaa painamalla paneelin oikeassa yläreunassa sijaitsevaa nappia. Sisältö saadaan takaisin näkyviin painamalla nappia uudelleen.

Jotta tukipyyntöä pystyy muokkaamaan, on kirjautuneen käyttäjän oltava tukipyynnön käsittelijä, koska jos tietoja muutetaan on hyvä jäädä merkintä siitä, kuka viimeksi on tukipyyntöä käsitellyt. Tukipyyntö-sivun latautuessa järjestelmä tarkastaa tukipyyntöön merkityn käsittelijän ja kirjautuneen käyttäjän tietoja keskenään. Jos käsittelijä ja kirjautunut käyttäjä eivät täsmää keskenään, järjestelmä ei anna muokata tukipyynnön tietoja muutamaa poikkeusta lukuun ottamatta. Toivottiin, jotta käsittelijä-, virheluokka- ja kategoriatietoa voisi muokata ilman, että tarvitsee olla käsittelijä. Kaikkiin muihin toimintoihin täytyy käyttäjän olla tukipyynnön käsittelijä. Mikäli käyttäjä ei ole tukipyynnön käsittelijä, voi käyttäjä ottaa tukipyynnön käsittelyyn klikkaamalla oikeassa alareunassa olevaa ota käsittelyyn-nappia. Täl-

lön järjestelmä muuttaa tukipyynnön käsittelijätiedon automaattisesti kirjautuneeksi käyttäjäksi. Toinen vaihtoehto on muuttaa käsittelijä itse vaihtamalla käsittelijä Perustiedot-paneelistä manuaalisesti omalle nimelle.

Tukipyyntö-sivulla Liite-paneelistä pystyy avaamaan liitteen. Kuvat avautuvat niiden nimeä klikkaamalla liite listan alapuolelle ja kuvan sulkeminen tapahtuu klikkaamalla Liite-paneelin oikeassa yläreunassa sijaitsevaa silmä-ikonia. PDF-tiedostot avautuvat selaimissa uuteen välilehteen. Jos tukipyynnön liite ei ole kuva tai PDF-tiedosto, järjestelmä antaa tiedoston tallennus-dialogin, jolloin sen voi ladata omalle laitteelle ja avata sen ohjelmalla, jolla se on tarkoitettu avattavaksi. Liite-paneelissa on myös mahdollista ladata tiedosto sen tyypistä riippumatta. Liite listassa on liitteen kohdalla lataus-ikoni, jolla tiedoston lataaminen on mahdollista.



Kuvio 22. Kuva tukipyynnön Liite-paneelistä jossa tukipyynnön liitekuva avattuna.

Tapahtumat-paneeli on yksi tärkein työkalu tukipyynnön käsittelyssä, koska sillä voidaan lähettää asiakkaalle lisäkysymyksiä ja tämän avulla voidaan myös sisäisesti keskustella yrityksen sisällä. Tapahtuma listassa näkyy erikseen yrityksen sisäiset

viestit, joita asiakas ei pääse näkemään ja ulkoiset viestit, jotka on tarkoitettu asiakkaan kanssa keskustelemiseen. Asiakasviestin voi lähettää vain tukipyyntöön merkitty käsittelijä.

Tapahtumalistassa nähdään myös tukipyynnön historia merkitty tapahtumina. Tietystä toiminnoista järjestelmä kirjaa tietokantaan merkinnän tästä tapahtumasta, joka tulee näkymään aina sen tukipyynnön tapahtumissa. Tapahtumat, joista järjestelmä kirjaa tapahtuman tietokantaan, ovat tilan muuttuminen, käsittelijän muuttuminen, tyyppin muuttuminen, ratkaisun muuttuminen, liitteen lisäys, liitteen poisto, yhteyshenkilön lisäys, yhteyshenkilön poisto, virheluokan muuttuminen ja sähköpostin lähettämisestä asiakkaalle.

Tapahtumalistan yläpuolella on tekstikenttä, johon haluamansa viestin voi kirjoittaa. Tekstikentän alapuolella on kolme nappia: asiakasviesti, jolla lähetetään ulkoinen viesti eli asiakkaalle tarkoitettu viesti, sisäinen viesti eli yrityksen sisäiseen käyttöön tarkoitettu viesti, vain viestit, jolla voidaan piilottaa tapahtumat, jotka eivät ole viestejä. Tällä saadaan helpotettua pelkkien viestien lukemista tapahtumalistasta.

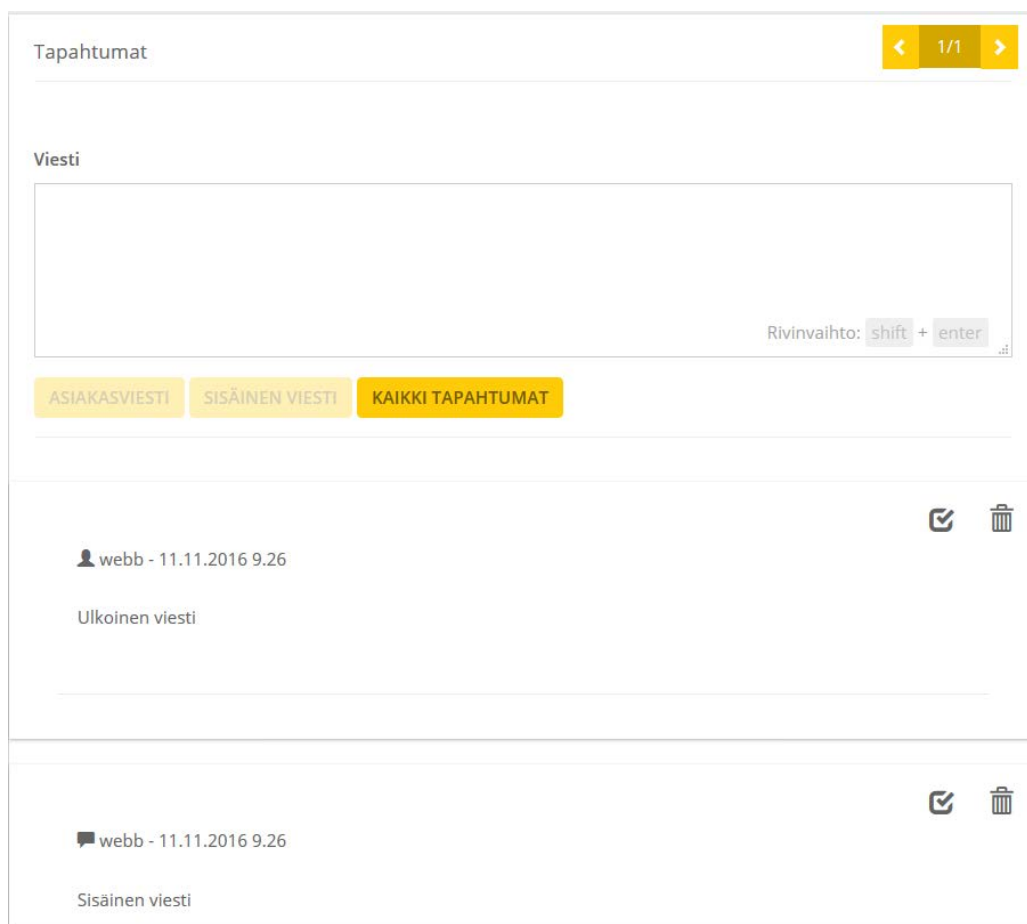
Tapahtumalista on sivutettu ja näyttää oletuksena viisi viimeisintä tapahtumaa. Tapahtumat-paneelin oikealla yläkulmassa on nappi, josta sivuja voidaan vaihtaa. Viestien kohdalla tapahtumalistalla on oikealla yläkulmassa ruksi-ikoni, josta valittu viesti voidaan merkata ratkaisuksi ja roskakori-ikoni, josta valittu viesti voidaan poistaa. Jokaisen tapahtuman kohdalla on vasemmalla oma ikoninsa, jotta ne erottaa toisistaan ja jokainen ikoni tarkoittaa tiettyä tapahtumaa. Tämä on toteutettu käyttämällä AngularJS:n `ng-class`-direktiiviä. Tietokannasta tulevilla tapahtumilla on kaikilla oma tyyppi-arvo numerona, jotta järjestelmä tunnistaisi eri tapahtumat. Kaikille tapahtuma tyyppin arvoille on liitetty `ng-class`-direktiivin sisälle oma ikoninsa.

```

<span class="glyphicon" data-ng-class="{ 'glyphicon-user': event.Event_type === 1,
'glyphicon-comment': event.Event_type === 2,
'glyphicon-tasks': event.Event_type === 3,
'glyphicon-transfer': event.Event_type === 4,
'glyphicon-eye-open': event.Event_type === 5,
'glyphicon glyphicon-paperclip': event.Event_type === 8,
'glyphicon glyphicon-trash': event.Event_type === 9,
'fa fa-user-plus': event.Event_type === 10,
'fa fa-user-times': event.Event_type === 11,
'glyphicon glyphicon-ok': event.Event_type === 12,
'glyphicon glyphicon-envelope': event.Event_type === 13,
'glyphicon glyphicon-dashboard': event.Event_type === 14 } ">

```

Kuvio 23. AngularJS:n ng-class-direktiivin käyttöä koodissa



Kuvio 24. Tukipyyntö Tapahtumat-paneeli

Tukipyyntö-sivun oikealla alakulmassa on tallenna-nappi, josta tukipyynnön muutokset voidaan tallentaa. Jos avattu tukipyyntö on merkitty kirjautuneelle henkilölle

käsittelyyn, tallenna-napin vieressä on myös merkitse valmiiksi- nappi, josta avattu tukipyyntö voidaan merkitä valmiiksi.

6 YHTEENVETO

Projektin tavoitteena oli rakentaa toimiva käyttöliittymä helpdesk-järjestelmälle käyttäen AngularJS-ohjelmistokehystä sekä muita ohjelmointiteknologioita. Olemme käyneet työssä läpi työn toteutukselle keskeisiä ohjelmointiteknologioiden osia ja työ sisältää runsaasti visuaalisia esimerkkejä käytännön toteutuksista. Työmme antaa kuvan siitä, kuinka voidaan luoda tehokkaat toimivat sivut hyvällä arkkitehtuuriratkaisulla useiden eri ohjelmointiteknologioiden ominaisuuksia hyväksi käyttämällä.

Projektin tutkimusongelmat saatiin ratkaistua käytettävissä olevilla teknologioilla. Tukikeskus-sivulla on saatu esitettyä aiheelliset, kooltaan rajatut tukipyyntölistat sekä erinäiset mittarit tukipyyntöjen kehityksestä AngularJS-direktiiveinä rakennetuilta widgeteilta.

Tukipyyntö-listalla ilmenneet tutkimusongelmat, kuten tietomäärän rajaaminen verkkoliikenteessä saatiin ratkaistua sivutetuin hauin. SQL-kyselyt on rakennettu siten, että vaikka kokonaistulosjoukko sisältäisi tuhansia rivejä, haetaan käyttöliittymällä asetetun sivun ja sivun koon mukaisesti pienempi tulosjoukko varsinaisen tulosjoukon oikeasta kohdasta. Tukipyyntölistan alustuksessa ilmennyt ongelma AngularJS:n \$http-palvelun kutsujen asynkronisuudesta ratkaistiin AngularJS:n \$q-palvelua käyttämällä. Tämän palvelun avulla saadaan ensin suoritettua asynkronisesti joukko metodeja, ja vasta näiden valmistuttua jatketaan ohjelmakoodissa eteenpäin.

Uusi tukipyyntö-sivulla tutkimusongelmat ratkaistiin lähinnä asettelemalla tietoja paneeleihin ja paneelit aseteltiin sivulle yleisimpään täyttöjärjestykseen. Tukipyyntö-sivulla tutkimusongelmat, kuten tiedon piilottaminen ratkaistiin AngularJS:n ng-hide- ja ng-show-direktiivin avulla. Kuvan esittäminen ratkaistiin avaamalla pdf-tiedosto erikseen uuteen välilehteen ja kuva esitetään Liite-paneelissa. Kaikki muut liitteet ladataan koneelle. Tapahtumat ratkaistiin sivuttamalla tietoja ja esittämällä tapahtumat listana Tapahtumat-paneelissa.

Vaikka työ saatiin valmiiksi ja sivusto toimii moitteettomasti, voidaan jälkikäteen tarkastellessa huomata muutamia korjattavia asioita. Listasivulla hakumetodit sisältävät suuren määrän parametreja, mikä on haitallista kun pyritään pitämään tiedon määrä verkkoliikenteessä mahdollisimman pienenä. Tähän voisi ratkaisuksi käyttää muutamia ennalta määriteltyjä rajauskokoelmia. Hakukriteerien ollessa täysin vapaasti valittavissa emme keksineet parempaa ratkaisua. Listahakuja varten olisi myös voitu toteuttaa tallennetut SQL-proseduurit, jotka ottaisivat vastaan tarvittavat muuttujat, niin olisi välttytty pitkien ja monimutkaisten SQL-kyselyjen muodostamiselta bisneslogiikan luokissa.

Opinnäytetyömme oli aiheena haastava ja mielenkiintoinen. Haastavuutta lisäsi huomattavasti se, että sovelluskokonaisuus sisälsi useita eri kerroksia ja osia. Ohjelmointikielten itsenäinen opiskelu varsinaisen työskentelyn ohella oli myös haastavaa ja lisäsi työhön kuluvaan aikaa.

Käyttöliittymä on ollut Lemonsoft Oy:llä testikäytössä jo kuukausia vanhan Silverlight-käyttöliittymän ohella ja se on saanut positiivista palautetta. Lähitulevaisuudessa edellinen Silverlight-käyttöliittymä poistetaan käytöstä kokonaan. Silverlight on Microsoftin kehittämä Adobe Flashin tyyppinen web-ohjelmointiympäristö.

Projekti onnistui hyvin. Asiakas sai toimivan tuotteen aikataulun mukaisesti ja ohjelmointitaitomme kehittyivät erittäin paljon projektin aikana. Olemme myös todennäköisesti molemmat työllistymässä tämän työn toimeksiantajalle koulutuksemme valmistuttua.

LÄHTEET

AngularJS.org 2016a. Developer guide: What are Directives?. Viitattu 13.11.2016. <https://docs.angularjs.org/guide/directive>

AngularJS.org 2016b. Developer guide: ngIf. Viitattu 13.11.2016. <https://docs.angularjs.org/api/ng/directive/ngIf>

AngularJS.org 2016c. Developer guide: \$q. Viitattu 13.11.2016. [https://docs.angularjs.org/api/ng/service/\\$q](https://docs.angularjs.org/api/ng/service/$q)

AngularJS.org 2016d. Developer guide: \$http. Viitattu 13.11.2016. [https://docs.angularjs.org/api/ng/service/\\$http](https://docs.angularjs.org/api/ng/service/$http)

AngularJS.org 2016e. Developer guide: ngclass. Viitattu 14.11.2016. <https://docs.angularjs.org/api/ng/directive/ngClass>

ASP.NET Web API dokumentaatio 2016 ASP.NET Web API. Viitattu 31.10.2016 [https://msdn.microsoft.com/en-us/library/hh833994\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/hh833994(v=vs.108).aspx)

Automapper dokumentaatio 2016 Home. Viitattu 25.10.2016 <https://github.com/AutoMapper/AutoMapper/wiki>

Bootstrap dokumentaatio 2016 Viitattu 31.10.2016 <https://getbootstrap.com/>

Highcharts dokumentaatio 2016 WHAT IS HIGHCHARTS?. Viitattu 31.10.2016 <http://www.highcharts.com/products/highcharts>

Lemonsoft DevHelp2014. Viitattu 14.11.2106

Seshadri S & Green B. 2014. AngularJS: Up And Running . O'Reilly.

Tutorialspoint.com 2016a What is MVC?. Viitattu 24.10.2016 https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

Tutorialspoint.com 2016b MVC Components. Viitattu 24.10.2016 https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

TypeScript dokumentaatio 2016 Introduction. Viitattu 25.10.2016 <https://github.com/Microsoft/TypeScript/blob/master/doc/spec.md>