

Firstbeat Hyvinvointianalyysin hallintajärjestelmän päivitys

Arttu Isopahkala

Opinnäytetyö

Huhtikuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), mediatekniikan tutkinto-ohjelma

Tekijä(t) Isopahkala, Arttu	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Huhtikuu 2017
	Sivumäärä 43	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi Firstbeat Hyvinvointianalyysin hallintajärjestelmän päivitys		
Tutkinto-ohjelma Mediatekniikka		
Työn ohjaaja(t) Manninen, Pasi		
Toimeksiantaja(t) Firstbeat Technologies Oy		
Tiivistelmä <p>Opinnäytetyö tehtiin hyvinvoinnin, huippu-urheilun ja kuluttajatuotteiden kanssa toimivalle Firstbeat Technologies Oy:lle. Firstbeat Hyvinvointianalyysin hallintajärjestelmä (engl. FLA Admin Client) on tarkoitettu asiakkaiden, erilaisten käyttäjätilien ja logistiikkakeskusten hallinnointiin. Järjestelmässä asetetaan mm. tietoja ja erilaisia oikeuksia tileille. Vanhempi järjestelmä oli vaikeaa käynnistää ja jotkin toiminnot olivat sekavia. Tästä syystä luotiin uusi web-ympäristössä toimiva hallintajärjestelmä, jonne järjestelmänvalvojat pääsivät kirjautumaan helposti vain verkkoselainta käyttäen. Päivityksen tarkoitus oli helpottaa Hyvinvointianalyysiin liittyvää hallinnointia niin, että vanhasta hallintajärjestelmästä voitiin luopua.</p> <p>Uudesta hallintajärjestelmästä luotiin käyttöliittymäsuunnitelmat ja käyttötapauskuvaukset, joiden perusteella järjestelmää alettiin kehittää. Hallintajärjestelmä toteutettiin Eclipse-kehitysympäristöön ladatun GWT-lisäosan avulla. GWT (Google Web Toolkit) on kehitysohjelma, joka yksinkertaisuudessaan kääntää Java-koodin JavaScriptiksi web-ympäristöön. Näin ollen Java-kieleen tottunut ohjelmoija pystyy luomaan laajoja ja dynaamisia web-sovelluksia tutulla ohjelmointikielellä. Järjestelmän kehityksessä käytettiin MVP-arkkitehtuurimallia, jonka avulla tiedostot ovat organisoitu omaan osa-alueeseen liitettyyn tietoon. Työ kertoo kuinka Java-kielellä voidaan toteuttaa monipuolinen REST-rajapintaa käyttävä sovellus web-ympäristöön ilman, että tarvitaan syvempää tietämystä erilaisiin selaimiin liittyvistä ominaisuuksista kuten XMLHttpRequest:ista tai JavaScriptista.</p> <p>Uuden hallintajärjestelmän valmistuttua hallinnointi helpottui ja vanhan järjestelmän käyttö voitiin lopettaa. Lisäksi käyttöliittymästä tehtiin käyttäjäystävällisempi ja yksinkertaisempi. Uusi hallintajärjestelmä kommunikoi paremmin käyttäjän kanssa. Dialogit ja ohjetekstit kertovat onnistuuko tehty toiminto tai haluaako käyttäjä varmasti suorittaa jonkin operaation. Järjestelmä valmistui ajallaan ja siihen saatiin sisällettyä kaikki tärkeimmät ja halutut ominaisuudet.</p>		
Avainsanat (asiasanat) Google Web Toolkit, MVP-arkkitehtuurimalli, web-sovelluksen kehitys		
Muut tiedot		

Author(s) Isopahkala, Arttu	Type of publication Bachelor's thesis	Date April 2017
	Number of pages 43	Language of publication: Finnish
Permission for web publication: x		
Title of publication Upgrade of Firstbeat Lifestyle Assessment management system		
Degree programme Media Engineering		
Supervisor(s) Manninen, Pasi		
Assigned by Firstbeat Technologies Oy		
Abstract <p>The thesis was assigned by Firstbeat Technologies Oy which works with heart rate variability based solutions for sports and fitness, health and occupational well-being. The management system for Firstbeat Lifestyle Assessment is created for managing customer accounts, different kind of user accounts and logistic centers. The former management system was difficult to start, and some functions were messy to understand. For this reason, a new management system working in web environment was created so system administrators would be able to enter the system easily using web browser only. The goal of the upgrade was to ease the management of customers and logistic centers so that the old management system would not be needed anymore.</p> <p>The development of the new management system started after user interface plans and use case descriptions were created. The management system was implemented with Eclipse development environment and its GWT plugin. GWT (Google Web Toolkit) is development toolkit which in its simplicity compiles Java code to JavaScript for web environment, which allows the developer to use the well-known Java programming language to build comprehensive and dynamic web applications. Code files were organized by the section of information using MVP architecture pattern. The final product shows how the high-performance application using REST interface can be created to web environment without deeper knowledge of browser quirks, XMLHttpRequest or JavaScript.</p> <p>After the new management system was published, management became easier and use of the old management system could be finished. Also, the user interface is now more user friendly and simpler. The new management system gives more information to a user about various operations. Dialogs and info messages inform and confirm if an operation was successfully completed or if the user really wants to execute it. The management system was finished in time including all most important functions planned in the first place.</p>		
Keywords/tags (subjects) Google Web Toolkit, MVP architecture pattern, Web application development		
Miscellaneous		

Sisältö

1	Työn lähtökohdat	4
1.1	Toimeksiantaja	4
1.2	Tehtävä ja tavoitteet	4
2	Käytetyt tekniikat ja työkalut.....	6
2.1	Google Web Toolkit.....	6
2.2	MVP-arkkitehtuurimalli	7
2.3	Eclipse ja Photoshop.....	9
3	Työn suunnittelu	10
3.1	Vaatimusmäärittely	10
3.2	Käyttöliittymäkuvat	10
3.3	Käyttötapauskuvaukset	11
4	Järjestelmän ominaisuudet.....	11
4.1	Kirjautumissivu	11
4.2	Asiakas- ja käyttäjätilit.....	12
4.2.1	Pääsivun ominaisuudet.....	13
4.2.2	Toimistot-dialogit	15
4.2.3	Koodit-dialogit	17
4.3	Centerien hallinta	19
4.4	Järjestelmänvalvojat.....	21
5	Työn toteutus	23
5.1	Rajapintakutsut	23
5.2	Käyttöliittymä	24
5.3	Kustomoidut komponentit	25
5.3.1	FbtGrid	25
5.3.2	FbtDateBox	27

	2
5.4 Dialogit.....	28
6 Tulokset	30
6.1 Tulos tavoitteisiin nähden	30
6.2 Vertailu vanhaan hallintajärjestelmään	31
6.3 Työn hyödyt yritykselle	33
7 Pohdinta.....	33
Lähteet	35
Liitteet	36

Kuviot

Kuvio 1. GWT:n käytön osuus Java-kehitystyökaluista	7
Kuvio 2. MVP-arkkitehtuurimallin toimintaperiaate.....	8
Kuvio 3. Kirjautumissivun sisältö	12
Kuvio 4. Käyttöliittymäkuva hallintajärjestelmän päänäköymästä.....	13
Kuvio 5. Asiakastilin muokkaamiseen ja luomiseen käytetty dialogi.....	14
Kuvio 6. Käyttäjätili-dialogin käyttöliittymäsuunnitelma.....	15
Kuvio 7. Toimistot-dialogi.....	16
Kuvio 8. Muokkaa toimistoa -dialogi.....	16
Kuvio 9. Koodit-dialogi	17
Kuvio 10. Luo tilauskoodeja -dialogi.....	18
Kuvio 11. Tutki tilauskoodeja -dialogi	18
Kuvio 12. Käyttöliittymäkuva hallintajärjestelmän Centerit-välilehdestä	19
Kuvio 13. Centerien muokkaamiseen ja luomiseen käytetty dialogi	20
Kuvio 14. Center-käyttäjien muokkaamiseen ja luomiseen käytetty dialogi.....	21
Kuvio 15. Käyttöliittymäkuva hallintajärjestelmän Hallinto-välilehdestä.....	22
Kuvio 16. Järjestelmänvalvojan muokkaamiseen ja luomiseen käytetty dialogi	22
Kuvio 17. Tiedon hakeminen tietokannasta ja sen kulku näkymään	24
Kuvio 18. FbtGrid luotu käyttäen "MULTISELECT_WITH_CTRL_OR_SHIFT"- enumeraatiota.....	26
Kuvio 19. FbtDateBox-kalenteri, jossa rasti päivämäärän tyhjentämistä varten	27

1 Työn lähtökohdat

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantaja Firstbeat Technologies Oy (jäljempänä Firstbeat) on vuonna 2002 perustettu kasvava jyvaskyläläinen yritys, joka toimii hyvinvoinnin, huippu-urheilun ja kuluttajatuotteiden parissa. Yrityksen kehittämä sydämen sykevälialanalyysi auttaa parantamaan suorituskykyä ja hyvinvointia. Monitieteiseen tutkimukseen perustuva analyysi tuottaa henkilökohtaista tietoa stressistä, palautumisesta ja liikunnan vaikutuksista. (Yritys 2017.)

Firstbeatin tuotteita hyödynnetään monissa työhyvinvoinnin palveluissa. Niitä käytetään myös huippu-urheilussa valmennuksen apuvälineenä. Lisäksi yritys tekee yhteistyötä kuluttajatuotteiden valmistajien kanssa, jotka sisällyttävät Firstbeatin kehittämää teknologiaa esimerkiksi sykemittareihin.

Jyvaskylän toimipisteessä työskentelee noin 80 työntekijää eri tehtävissä. Lisäksi Firstbeatilla on muutaman työntekijän toimipiste Espoossa. Firstbeat sai valtakunnallisen yrittäjäpalkinnon 2016. (Valtakunnallinen yrittäjäpalkinto Firstbeat Oy:lle 2016.)

1.2 Tehtävä ja tavoitteet

Hyvinvointianalyysin hallintajärjestelmä on suunniteltu Firstbeatin Hyvinvointianalyysi-palvelua varten. Hyvinvointianalyysi on yksityisille kuluttajille ja ryhmille suunnattu palvelu, jossa henkilön sykeväliä mitataan kolmen vuorokauden ajan. Mittauksesta saadaan paljon tietoa mm. unen laadusta, stressitasoista, palautumisesta ja liikunnan vaikutuksista. Henkilö pitää mittauksen ajan päiväkirjaa työajoista, unijaksoista ja arkirutiineista, minkä perusteella analyysi kertoo tarkempaa tietoa hyvinvoinnin tilasta erilaisien tilanteiden ajalta. Mittauksen jälkeen henkilö saa raportin mittauksesta, jonka perusteella asiantuntija antaa palautetta mitatulle ja vinkkejä hyvinvoinnin parantamiseen. Hyvinvointianalyysin tilaamiseen ja tekoon on erilaisia variaatioita mm. laitteiden tilauksen, raporttien ja palautteenannon suhteen. Mittauksen voi tehdä joko omalla laitteella tai Firstbeat

Centerin (myöhemmin Center) avulla. Center on logistiikkakeskus, jossa vastaanotetaan mittalaitetilauksia ja puretaan data valmistuneista mittauksista.

Ennen opinnäytetyön aloitusta yrityksessä käytettiin Java-kielellä ohjelmoitua jo vuosia vanhaa hallintajärjestelmää. Järjestelmä oli JAR-tiedosto, joka avattiin BAT-tiedoston tai komentorivin kautta. Enemmistö ajoi tiedostoa yleisen verkkolevyn kautta. Työntekijöiltä oli tullut jonkin verran palautetta, että järjestelmää on vaikea käyttää sen sekavuuden takia ja että siinä ilmeni välillä vikoja, kuten skaalautumista eri näyttöpäätteille ja toimintoja, jotka eivät toimineet odotetulla tavalla.

Tehtäväksi annettiin järjestelmän päivitys nykyaikaisempaan malliin, joka tehtäisiin alusta asti uusiksi Google Web Toolkitia käyttäen. Uuden järjestelmän täytyi toimia web-ympäristössä. Näin järjestelmään päästäisiin kirjautumaan miltä tahansa laitteelta vain verkkoselainta käyttäen. Tavoitteena oli tehdä käyttöliittymästä selkeämpi ja käyttäjäystävällinen. Toimintojen tuli toimia paremmin ja ilmoittaa käyttäjälle selkeämmin, mitä ohjelmassa tapahtuu. Uudella hallintajärjestelmällä piti pystyä hallitsemaan seuraavia asioita:

- Asiakastilit
- Asiakkaille kuuluvat käyttäjätilit
- Centerit
- Centerien työntekijöiden käyttäjätilit
- Järjestelmänvalvojien käyttäjätilit

Työn tehtäviin kuului uuden järjestelmän suunnittelu tarkasti aina käyttöliittymäkuvia myöten. Työstä tuli kirjoittaa käyttötapauskuvaukset testausta varten sekä miettiä toiminnot, jotka vanhasta järjestelmästä toteutetaan uuteen. Jo alussa tiedettiin, että järjestelmän vaatimukset voivat tulla päivittymään vielä työn toteutusvaiheessa, mutta kun ensimmäinen suunnitelma hallintajärjestelmästä on valmistunut, aloitetaan kuitenkin sen ohjelmointi. Järjestelmää voidaan päivittää myöhemmin uusien vaatimusten mukaisesti. Sovellus tuli voida ottaa käyttöön sellaisenaan niin ettei vanhaa järjestelmää enää tarvita.

2 Käytetyt tekniikat ja työkalut

2.1 Google Web Toolkit

Google Web Toolkit (myöhemmin GWT) on kehitystyökalu, jolla voidaan rakentaa ja optimoida monimutkaisia selainpohjaisia sovelluksia. Sen tavoite on mahdollistaa korkean suorituskyvyn omaavien web-sovellusten tehokas kehitys ilman, että kehittäjän tarvitsee tietää syvemmin selaimen omista ominaisuuksista kuten XMLHttpRequeststeista tai JavaScriptistä. GWT:tä käytetään monissa Googlen tuotteissa. Tällaisia ovat mm. AdWords, AdSense, Flights, Hotel Finder, Offers, Wallet ja Blogger. GWT on avointa lähdekoodia, täysin ilmainen ja sitä käyttävät tuhannet kehittäjät ympäri maailman. (Overview 2017.)

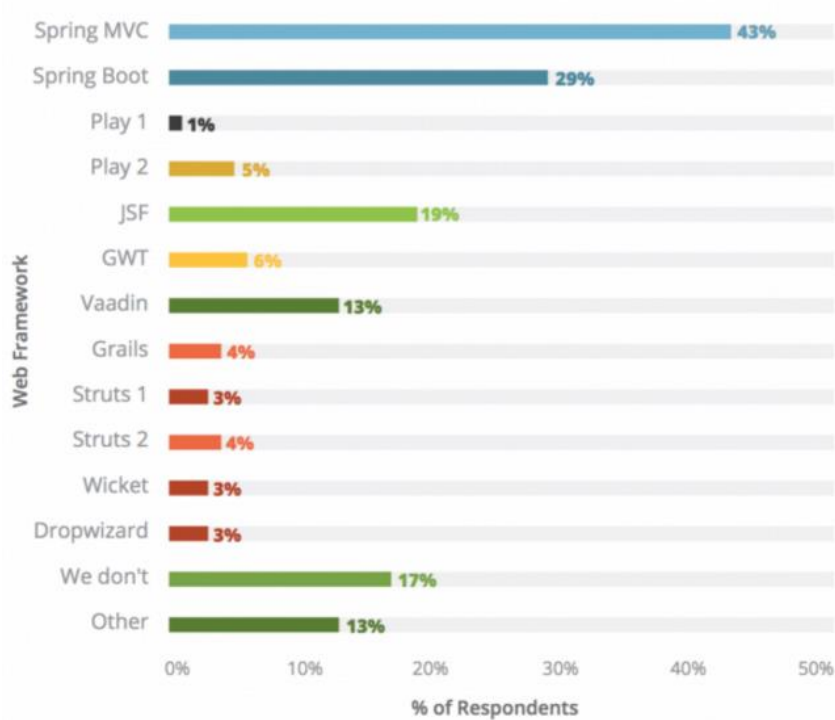
GWT:n peruseriaate on, että se kääntää Java-koodin JavaScriptiksi web-ympäristöön. GWT sisältää lisäosan Eclipseen, jota voidaan käyttää sovellusten kehittämiseen. Vuonna 2015 Eclipseä käytettiin kehitysympäristönä 72 prosentissa GWT-projekteista. (The Future of GWT Report 2015.)

GWT SDK sisältää perus Java API:t ja widgetit. Näiden avulla voidaan kirjoittaa AJAX-sovelluksia Java-koodilla ja kääntää optimoitu lähdekoodi JavaScriptiksi, joka toimii kaikilla selaimilla mukaan lukien mobiiliselaimet Androidilla ja iPhonella. Tällä tavalla kehitetyt AJAX-sovellukset ovat tuotteliaampia yhteisten käsitteiden kuten DOM-manipulaation ja XHR-viestinnän myötä. Kaikki, mitä voidaan tehdä selaimen DOM:illa ja JavaScriptillä, voidaan tehdä myös GWT:n avulla. Tämä mahdollistaa valmiiden käyttöliittymäelementtien muokkaamisen tai kokonaan uusien tekemisen ja kustomoimisen. (Overview 2017.)

GWT:n virheenkorjaus toimii samalla tavalla kuin normaaleissa JavaScriptillä ohjelmoiduissa web-sovelluksissa. Käytössä on tuttu kaava: ensin muokataan, sitten päivitetään ja katsotaan. Selaimen kehittäjäkonsolissa voidaan katsoa lokimerkintöjä, asettaa pysäytyspisteitä ja tutkia muuttujia. Samaan aikaan voi käyttää hyödyksi ohjelmointiympäristön omia virheenkorjaustyökaluja Javalle.

Java-kehitystyökaluista GWT ei kuitenkaan ole suosituin. Vuonna 2016 kuusi prosenttia Java-kehitystyökalujen käyttäjistä ilmoitti käyttävänsä tai käyttäneensä GWT:tä (ks. kuvio 1). Käyttäjämäärä on ollut hieman laskussa, koska vielä 2015 sitä

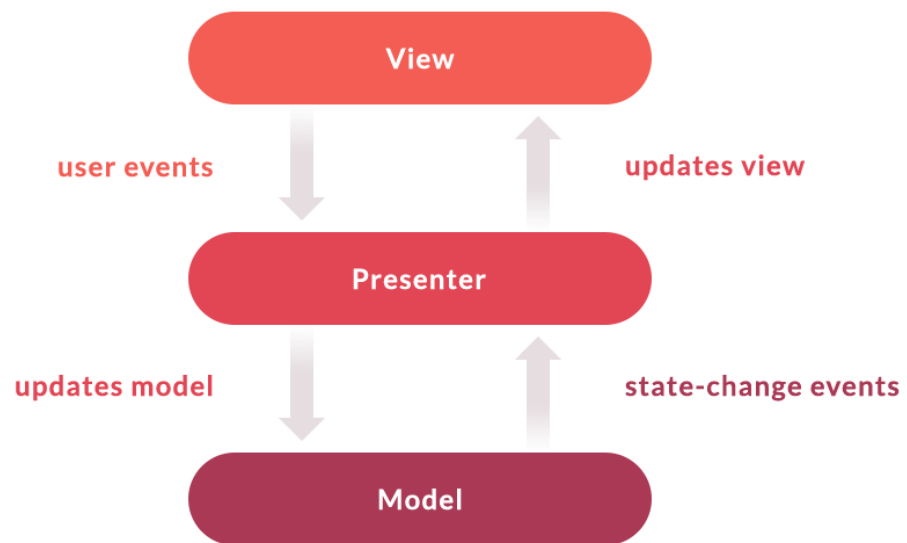
ilmoitti käyttävänsä 10 prosenttia käyttäjistä. Melko pieni prosenttiosuus johtuu ehkä siitä, että GWT:tä käytetään yleensä vain laajoihin liikemaailman web-sovelluksiin joihin se parhaiten sopii. (Maple 2016, The Future of GWT Report 2015.)



Kuvio 1. GWT:n käytön osuus Java-kehitystyökaluista (Maple 2016)

2.2 MVP-arkkitehtuurimalli

Malli-Näkymä-Esittäjä (engl. MVP – Model View Presenter) on IBM:n kehittämä malli-näkymä-arkkitehtuuri. MVP tarjoaa yleiskäyttöisen arkkitehtuurimallin käyttöliittymäkomponenttien sekä käyttöliittymäsovellusten kehitykseen. Kehitysmenetelmä sai alkunsa vuonna 1996 Mike Potelin toimesta. MVP perustuu vanhempaan MVC (Model View Controller)-arkkitehtuurimalliin ja siinä on paljon samoja piirteitä. Suurin ero on käyttäjien syötteiden käsittelyssä. MVP-arkkitehtuurimallissa näkymä vastaa käyttäjän syötteiden vastaanottamisesta (ks. kuvio 2), kun taas MVC-arkkitehtuurimallissa siitä vastaa ohjain (Controller). (Lehtinen 2011.)



Kuvio 2. MVP-arkkitehtuurimallin toimintaperiaate (Ziomacki 2016)

Suurien sovelluksien toteuttamisessa on aina vaikeutensa useiden kehittäjien ohjelmoidessa samanaikaisesti samaa lähdekoodia. Ominaisuuksien ja toimintojen yhtäaikaisten ohjelmointi voi helposti muuttua sekavaksi. Useista arkkitehtuurimalleista kuitenkin MVP toimii GWT:n kanssa parhaiten. Tähän on kaksi pääsyytä. Ensinnäkin MVP erittelee kehitystä eri osa-alueisiin, mikä mahdollistaa useiden kehittäjien yhtäaikaisten työskentelyn sovelluksen parissa. Toiseksi tämä malli minimoi testaukseen kulutettujen resurssien määrän, koska verkkoselaimien tuomia eriävyyksiä ei tarvitse huomioida. Lisäksi järjestelmättestaus voidaan toteuttaa lähinnä automaattisilla testitapauksilla, koska eri näkymissä hyödynnetään samoja esittäjään toteutettuja toiminnallisuuksia. Näin ollen riittää, kun kirjoittaa testitapauksia, jotka testaavat esittäjän toiminnallisuuksia. MVP-arkkitehtuurimallin mukaisen sovelluksen kehittäminen voi olla yksinkertaista ja helppoa, kun vain ymmärtää sen perusteet ja hyödyt. (Ramsdale 2010.)

MVP-arkkitehtuurimallissa näkymä (View) seuraa käyttäjän tekemiä toimintoja ja hallitsee niistä aiheutuvia tapahtumia. Esittäjään (Presenter) ohjelmoituja toiminnallisuuksia kutsutaan näkymän kautta. Kutsut tapahtuvat rajapinnan tai tapahtumanlaukaisujen avulla. Esimerkiksi kun käyttäjä painaa painiketta käyttöliittymässä, lähetetään kutsu rajapinnan kautta esittäjälle. Näin näkymä ei itse

toteuta logiikkaa. Esittäjän toiminnot tekevät tarvittavat muutokset malliin (Model). Malli sisältää esimerkiksi olion tiedot ja metodit. Esittäjään toteutettuja toiminnallisuuksia voidaan kutsua erilaisista käyttöliittymistä tai suoraan jonkin toisen ohjelman toimesta. Sen jälkeen esittäjä tekee malliin muutokset, jotka välitetään takaisin näkymään. Esittäjän avulla sovelluslogiikka ja tiedonhallinta on rajattu selkeäksi kokonaisuudeksi. (Lehtinen 2011.)

MVP-arkkitehtuurimallin käyttämisessä on monia hyötyjä. Rakenne on selkeä ja monikäyttöinen, koska sovelluksen osa-alueet sijoittuvat omiin tiedostoihin ja tekevät vain niille kuuluvia asioita. Tämä mahdollistaa ohjelman yhtäaikaisen kehittämisen usean kehittäjän toimesta helposti. Osa voi keskittyä näkymien suunnitteluun ja toteutukseen ja osa taas toiminnallisuuksien kehittämiseen. Esittäjien luomia rajapintoja voidaan käyttää sovelluksen eri kohdissa, mikä mahdollistaa koodin uusiokäytön. MVP-arkkitehtuurimalli mahdollistaa helpon testattavuuden. Jos virheitä syntyy, on helppoa löytää virheen aiheuttaja, koska esittäjän kapseloima sovelluslogiikka on rajattuna aina tiettyyn osaan järjestelmästä.

2.3 Eclipse ja Photoshop

Eclipse on ohjelmointiympäristö, joka tukee mm. Java-ohjelmointikieltä. Lisäksi tuettuja ohjelmointikieliä ovat esimerkiksi C, C++ ja PHP. IBM alkoi kehittää Eclipseä vuonna 1993, ja vuonna 2001 se julkaistiin avoimen lähdekoodin lisenssille. Opinnäytetyössä Eclipse valittiin ohjelmointiympäristöksi, koska GWT:n ainoa lisäosa oli suunniteltu juuri sitä varten. (About the Eclipse Foundation 2017.)

Opinnäytetyön graafinen suunnittelu tehtiin Adobe Photoshopia (myöhemmin Photoshop) käyttäen. Photoshop on Adobe Systemsin kehittämä kuvankäsittelyohjelma, joka on maailmanlaajuisesti kuuluisin ohjelma digitaalisten kuvien muokkauksessa. Ensimmäinen versio Photoshopista luotiin jo 1988 Thomas and John Knollin toimesta. Ensimmäisestä julkaisusta lähtien Photoshop on ollut yksi tunnetuimmista kuvankäsittelyohjelmista muun muassa sen helppouden ja monikäyttöisyyden takia. Photoshop on tarkoitettu lähinnä ammattikäyttöön, ja se tarjoaa monipuoliset mahdollisuudet digitaalikameralla otettujen kuvien muokkaamiseen tai grafiikan luomiseen esimerkiksi www-sivuille. Nykyään entistä useampi kuvankäsittelijä käyttää Photoshopia käyttöliittymien suunnitteluun.

(Photoshop Reigns Supreme: How the Software Has Maintained Market Dominance 2017.)

3 Työn suunnittelu

3.1 Vaatimusmäärittely

Ennen työn aloitusta oli tehtävä vaatimusmäärittely. Vaatimusmäärittelyn suunnittelussa täytyi miettiä seuraavia kysymyksiä: Mitä kaikkea hallintajärjestelmällä täytyy pystyä tekemään, millaisia muutoksia rajapintoihin sen toteuttaminen vaatii edelliseen hallintajärjestelmään verrattuna ja miten GWT:n tekniikkaa voi hyödyntää parhaiten? Tai mitä rajoitteita teknillisesti voi tulla vastaan? Nämä kysymykset mielessä alettiin tekemään listaa, jonka pohjalta myöhemmin rautalankamalleja ja viimeisteltyjä käyttöliittymäkuvia.

Vanhassa hallintajärjestelmässä oli joitain ominaisuuksia, jotka olivat tarpeettomia yrityksen kehityksen myötä. Näistä ominaisuuksista täytyi osa jättää pois. Suurimmat pois jääneet kokonaisuudet olivat kaksi välilehteä. Ensimmäisenä pois jäi ”Customizing” (ks. liite 1), jonka avulla pystyi hallitsemaan asiakkaalle näkyvän käyttöliittymän tyyliä. Toinen pois jäänyt välilehti oli ”Operation Statistics” (ks. liite 2), jota kautta näki palvelun tilastoja. Näiden lisäksi pois jätettiin erilaisia pienempiä ominaisuuksia, oikeuksia ja valintoja. Pois jääneet asiat ja niiden sisältö on lueteltu tarkemmin luvussa 6.2.

Vaatimusmäärittely muuttui pitkän prosessin, mutta alussa tehtiin sopivan kokoinen lista, joka käytiin palaverissa läpi muilla osastoilla työskentelevien henkilöiden kanssa. Sitä mukaan kun keskusteluja käytiin, syntyi myös uusia vaatimuksia, jotka vaikuttivat toteutukseen. Nämä täytyi kirjata ylös tulevaa varten.

3.2 Käyttöliittymäkuvat

Vaatimusmäärittelyn pohjalta tiedettiin, mitä käyttöliittymältä vaaditaan. Hallintajärjestelmän perusrakenne pysyi suhteellisen samanlaisena vanhaan järjestelmään nähden, koska se oli tarpeeksi järkevästi suunniteltu. Järjestelmässä tulisi olla kolme välilehteä, joiden kautta pystyisi hallitsemaan eri osioita. Yksi

välilehti on pääsivu, joka sisältää asiakas- sekä käyttäjätileihin liittyvät toiminnot. Toinen on Centerit, joka sisältää Centerien sekä niiden työntekijöiden hallinnan. Kolmas välilehti on hallinto, joka sisältää järjestelmänvalvojiin liittyvät toiminnot.

Käyttöliittymäkuvia suunniteltiin Photoshopin avulla. Värimaailma otettiin jo valmiista yrityksen SPORTS-hallintajärjestelmästä, joka oli hyvin yksinkertainen. Se sisälsi ainoastaan valkoista, vaaleansinistä ja harmaita värejä. Käytännössä kuviin sijoiteltiin komponentit uusilla tyyleillä suoraan vanhan järjestelmän käyttöliittymän perusteella, kuitenkin siten, että ne olisivat mahdollisimman käytännöllisesti ja siististi aseteltu. Koska osa komponenteista jätettiin pois, täytyi muutama sijoittelu miettiä uudelleen. Kuvia luotiin aidoilla komponenteilla, käyttäen Google Chromen Inspector-työkalua. Komponenttia muokattiin kevyesti, lisättiin uutta tekstiä ja tyylejä, jonka jälkeen siitä napattiin kuva ja siirrettiin Photoshopiin järjesteltäväksi.

3.3 Käyttötapauskuvaukset

Kun käyttöliittymäkuvat valmistuivat, tehtiin järjestelmästä käyttötapauskuvaukset (ks. liite 3) lähinnä testaajia varten. Käyttötapauskuvausten ensisijainen tehtävä oli kertoa, mitä järjestelmän toimintojen kuuluu tehdä. Esimerkiksi täytyi kertoa millaisia validointeja ja rajoitteita eri kentät sisältävät, ja mitä tapahtuu sen jälkeen, kun jokin toiminto on suoritettu. Näin testaajat pystyivät kirjoittamaan testitapaukset ja suorittamaan järjestelmälle tarkan testauksen.

Käyttötapauskuvaukset kirjoitettiin myös muita kehittäjiä varten. Mikäli työtä täytyisi jakaa esimerkiksi ajanpuutteen takia, toinen kehittäjä voisi jatkaa siitä, mihin toinen jäi, tai alkaa ohjelmoimaan jotain toista osaa järjestelmästä tietäen sen, mitä toimintojen tulee suorittaa. Käyttötapauskuvaukset kirjoitettiin englanniksi ja niihin sisällytettiin viittauksia suunniteltuihin käyttöliittymäkuviin.

4 Järjestelmän ominaisuudet

4.1 Kirjautumissivu

Hallintajärjestelmän kirjautumissivu (ks. kuvio 3) on yksinkertainen sivu, jonka avulla järjestelmään kirjaututaan. Käyttäjällä täytyy olla järjestelmänvalvojan oikeudet

päästäkseen kirjautumaan sisään. Kirjautumissivulla sisältää kentät joihin kirjoitetaan käyttäjänimi ja salasana. Lisäasetuksina voidaan valita hallintajärjestelmässä käytetty kieli ja se, että muistetaanko käyttäjänimi. Mikäli käyttäjä on unohtanut salasanansa, voidaan uusi generoitu salasana lähettää käyttäjän sähköpostiin syöttämällä käyttäjänimi ”Unohditko salasanasi?”-painikkeen takaa löytyvään kenttään. Myös hallintajärjestelmän versionumero näytetään kirjautumissivun alareunassa.



FLA Admin Client

Give your credentials and log in

Language

Username

Password

Remember my username

Kuvio 3. Kirjautumissivun sisältö

4.2 Asiakas- ja käyttäjätilit

Asiakastilit ovat Firstbeatin asiakkaille luotuja tilejä, joille asetetaan erilaisia oikeuksia ja tietoja asiakkaasta riippuen. Asiakkaalle kuuluu käyttäjätilejä, joista yksi käyttäjäryhmä on loppukäyttäjät (End-user). Loppukäyttäjätilit sisältävät Hyvinvointianalyysi-mittauksen tekevien henkilöiden profiilien tiedot. Toinen käyttäjätilyhmä on asiantuntijat (Pro), jotka yleensä hoitavat asiakkaan mittauksen analysoinnin, raporttien luonnin ja palautteenannon. Korkein käyttäjäryhmä on asiakastilin ylläpitäjät (Customer Admin), jotka voivat myös hoitaa edellä mainitut

asiat. Asiakastilin ylläpitäjällä on pääsy Hyvinvointianalyysi-ohjelman ylläpitäjille tarkoitettuun osioon, jossa ylläpitäjä voi itse määrittää joitain asiakkaalle kuuluvia ominaisuuksia. Asiantuntijakäyttäjille ja asiakastilin ylläpitäjille annetaan tunnukset, joilla päästään kirjautumaan Firstbeatin Hyvinvointianalyysi-ohjelmaan.

4.2.1 Pääsivun ominaisuudet

Hallintajärjestelmän päänäkymässä (ks. kuvio 4) on listattuna asiakastilit ja käyttäjätilit. Välilehden yhteiset ominaisuudet on esitetty ylhäällä omassa tilassaan. Niihin kuuluu kaksi hakutyökalua: asiakastilien etsiminen käyttäjänimen ja tilauskoodin perusteella. Lisäksi ”Käyttömääräraportti”-painikkeesta aukeaa dialogi, jolla voidaan ladata Excel-tiedosto, jossa näkyvät valitun vuoden asiakastilien myyntitilatot. ”Päivitä”-painikkeella voidaan päivittää välilehti alkutilaansa.

The screenshot shows the FLA Admin Client interface. At the top, there are navigation tabs for 'Main', 'Centers', and 'Administration'. Below this is a 'General' section with search filters for 'Search account by user' and 'Search account by order code'. The main area is divided into two panels: 'Accounts' and 'Users'. The 'Accounts' panel contains a table with columns for Name, Valid until, and Edit. The 'Users' panel contains a table with columns for Name, Email, Role, Office, and Edit. At the bottom, there are buttons for 'History', 'Report for invoicing', 'Account statistics', 'Create new account...', 'Move', 'Re-send credentials', and 'Create new user...'.

Name	Valid until	Edit
adasdasdadadada	08.02.2012	Edit...
Annun Testi 1		Edit...
Another Big customer	10.01.2012	Edit...
Antin testitili	13.08.3000	Edit...
Artun testi		Edit...
Artun testi 2		Edit...
Artun testi 3		Edit...
Artun testi 5		Edit...
Automated Tests #1	20.01.2022	Edit...
brand new	09.05.2016	Edit...
Center Test Account	03.05.2019	Edit...
Center Webshop Account	08.09.2019	Edit...
Emmin testitili	14.05.2020	Edit...
Eventizer	13.03.2011	Edit...
FeedbackSurveyTestCustomer	31.03.2018	Edit...
Fennia Online	19.05.2017	Edit...

Name	Email	Role	Office	Edit
1234, 123	arttu.isopahkala@firstbeat.cx	Customer Admin	Toimisto 2	Edit...
<zxcv, <xcz	xcvz@asd.fi	End-user	Toimisto 1	Edit...
ads, asd	asd@das.fi	End-user		Edit...
adsdas1234, asdasd1234	arttu.isopahkala@firstbeat.co	End-user		Edit...
afs@asd.sv	afs@asd.sv	End-user		Edit...
arttu-b2@firstbeat.com	arttu-b2@firstbeat.com	End-user		Edit...
arttu-b3@firstbeat.com	arttu-b3@firstbeat.com	End-user		Edit...
arttu-b4@firstbeat.com	arttu-b4@firstbeat.com	End-user		Edit...
arttu-c10@firstbeat.com	arttu-c10@firstbeat.com	End-user		Edit...
arttu-c1@firstbeat.com	arttu-c1@firstbeat.com	End-user		Edit...
arttu-c2@firstbeat.com	arttu-c2@firstbeat.com	End-user		Edit...
arttu-c3@firstbeat.com	arttu-c3@firstbeat.com	End-user		Edit...
arttu-c4@firstbeat.com	arttu-c4@firstbeat.com	End-user		Edit...
arttu-c5@firstbeat.com	arttu-c5@firstbeat.com	End-user		Edit...
arttu-c7@firstbeat.com	arttu-c7@firstbeat.com	End-user		Edit...

Kuvio 4. Käyttöliittymäkuva hallintajärjestelmän päänäkymästä

Asiakas- ja käyttäjätileille on omat hakukenttensä, joilla voidaan etsiä tietty tili nimen perusteella. Käyttäjän painaessa asiakastiliä listalta, ohjelma lataa sen sisältämät käyttäjätilit ja näyttää ne oikeanpuolisessa listassa. Asiakastilejä voidaan muokata tai poistaa painamalla ”Muokkaa”-painiketta valitun asiakastilin kohdalla, jolloin aukeaa dialogi (ks. kuvio 5), jota kautta voidaan myös luoda mm. laitetilauskoodeja asiakastilin käyttäjille.

The screenshot shows a 'Account details' dialog box with the following sections and fields:

- Account information:**
 - Account name*: Artun testi
 - Address: sadgg
 - Contact person: Arttul
 - Phone number: 34347773
 - Country: Finland
 - Manager: Arttu
 - Default Professional: arde
 - Notes: This is
 - Emails for automatic reporting: arttu.isopahkala@firstbeat.com
 - Offices: Manage...
- Account settings:**
 - Max Professional users*: 6
 - Free Professional users: 0
 - Show ectopic beats in Professional report
 - Ask permission to disclose information
 - Valid from: -
 - Valid until: -
- Billing information:**
 - Billable
 - Lower price for summary report
 - Billing customer id: 12
- Credits:**
 - Report credits: 5 (Add)
 - All time report credits: 6
 - Center credits: 2 (Add)
 - All time center credits: 314
 - Codes...
- Allowed services:**
 - Center: Firstbeat Center
 - Data export
 - Webshop
 - Appointments

Buttons: Delete, Save, Cancel

Kuvio 5. Asiakastilin muokkaamiseen ja luomiseen käytetty dialogi

Asiakastililistan alla on erilaisia painikkeita, joista osalla voidaan tutkia valitun asiakastilin tietoja tai suorittaa sille yleisiä toimintoja:

- "Asiakastilin tilastot"-dialogi näyttää nimensä mukaisesti tilastoja asiakastilin käyttäjätileistä sekä hyvinvointianalyseistä. Tilastoja voidaan rajata valitun aikavälin mukaan.
- "Luo uusi asiakastili"-dialogin kautta voidaan luoda uusi asiakastili. Tätä dialogia käytetään myös asiakastilien muokkaamiseen.
- "Raportti laskutukseen"-dialogissa voidaan ladata PDF-muotoinen laskutusraportti halutulta aikaväliltä.
- "Historia"-dialogissa näytetään valitun asiakastilin toimintahistoria.

Asiakastilin alla olevat käyttäjätilit näkyvät oikealla listassa. Uusi käyttäjätili voidaan luoda "Luo uusi käyttäjä"-painikkeen kautta. Käyttäjätilejä voidaan muokata tai

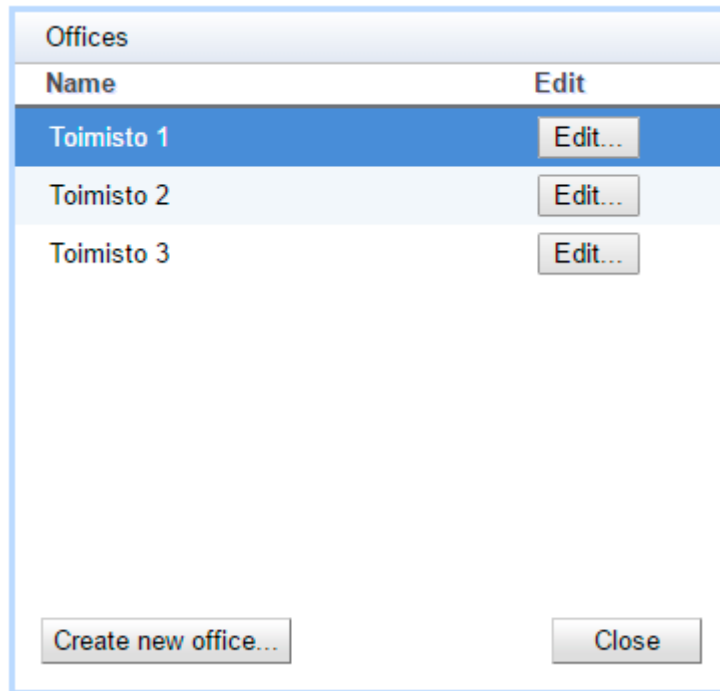
poistaa ”Muokkaa”-nappia painamalla käyttäjätilin riviltä, jolloin aukeaa tilin muokkaamiseen ja luomiseen käytetty dialogi (ks. kuvio 6).

Kuvio 6. Käyttäjätili-dialogin käyttöliittymäsuunnitelma

”Lähetä kirjautumistiedot”-painikkeella lähetetään uusi generoitu salasana kaikkien listalta valittujen asiantuntijakäyttäjien ja asiakastilin ylläpitäjien sähköpostiin. Listalta valittuja käyttäjätilejä voidaan siirtää muihin asiakastileihin painamalla ”Siirrä”-painiketta ja valitsemalla haluttu asiakastili. Käyttäjätilin muokkaamisdialogissa voidaan ladata lisäksi käyttäjän tekemät hyvinvointianalysimittaukset omalle tietokoneelle.

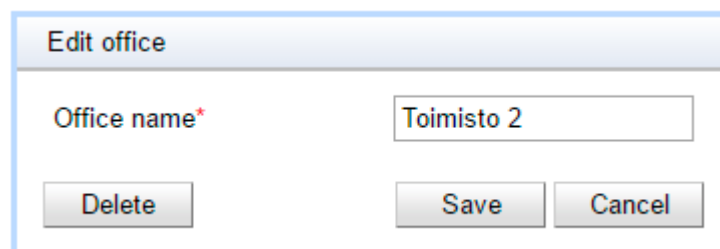
4.2.2 Toimistot-dialogit

Toimistoja käytetään asiakkaan laskutuksessa ja raporttien luonnista kuluviin kredittien tilastoinnissa. Koska asiakkaalla voi olla eri toimipaikkoja, täytyi käyttäjätileille pystyä asettamaan toimisto. Tätä varten suunniteltiin pieni dialogiryhmä, jolla toimistoja saattoi lisätä, poistaa ja muokata. Asiakastilin dialogissa käyttäjän painaessa ”Hallitse” aukeaa dialogi (ks. kuvio 7), johon on listattuna asiakaskohtaiset toimistot.



Kuvio 7. Toimistot-dialogi

Käyttäjä voi luoda dialogin kautta uuden toimiston painamalla "Luo uusi toimisto". Jo luotuja toimistoja pääsee muokkaamaan tai poistamaan painamalla "Muokkaa"-painiketta halutun toimiston kohdalta, jolloin aukeaa dialogi (ks. kuvio 8). Kun toimiston nimi on kirjoitettu dialogin tekstikenttään ja "Tallenna"-painiketta on painettu, toimisto tallentuu asiakastilille ja asettuu näkyväksi käyttäjätili-dialogin "Toimisto" alasvetovalikkoon.



Kuvio 8. Muokkaa toimistoa -dialogi

4.2.3 Koodit-dialogit

Järjestelmänvalvojat voivat luoda tilauskoodeja asiakastilin asiantuntijolle tai ryhmille. Loppukäyttäjä voi käyttää hänelle annetun koodin tilatessaan hyvinvointianalyysiä. Koodi syötetään tilauslomakkeeseen, joka sijaitsee omalla web-sivullaan. Asiakastilin dialogissa ”Koodit”-painiketta painamalla aukeaa uusi dialogi (ks. kuvio 9), johon on listattu tiedot luoduista tilauskoodeista.

Codes							
Created	Expires	Professional	Group	Language	Additional info	Used / Total	Details
02.02.2017	02.02.2018	Isopahkala, Arttu		Finnish	dasdas	0 / 1	Open...
31.01.2017	31.01.2018		Testi ryhmä 3000	Finnish	testi	0 / 3	Open...
17.11.2016	17.11.2017	testi1, Artun		Finnish	af	0 / 1	Open...
18.10.2016	19.10.2017	asd, Artturi		Finnish		0 / 4	Open...
18.10.2016	18.10.2017	iasdopas, Arttu		Finnish		0 / 6	Open...
18.10.2016	18.10.2017		Artun testiryhmä	Finnish	uusi2	0 / 1	Open...
18.10.2016	18.10.2017	iasdopas, Arttu		Finnish	uusi	0 / 1	Open...
13.10.2016	13.10.2017	asd, Artturi		Finnish		0 / 1	Open...
12.10.2016	12.10.2017	asd, Artturi		Finnish	th	0 / 1	Open...
11.10.2016	11.10.2017	iasdopas, Arttu		Finnish	aa	1 / 1	Open...

Create order codes Close

Kuvio 9. Koodit-dialogi

Tilauskoodeja pääsee luomaan painamalla ”Luo tilauskoodeja”-painiketta. Dialogissa (ks. kuvio 10) kysytään tilauskoodien lukumäärä, mihin ryhmään tai kenelle asiantuntijalle koodit kuuluvat, milloin koodit vanhenevat ja muita lisätietoja. Luodessaan tilauskoodit ohjelma lataa tilauskoodit valinnan mukaan tietokoneelle joko PDF- tai Excel-tiedostona. Tilauskoodien luomisessa käytettyä dialogia käytetään myös niiden muokkaamiseen.

Kuvio 10. Luo tilauskoodeja -dialogi

Tilauskoodeja pääsee tarkastelemaan ja muokkaamaan painamalla ”Aava”-painiketta koodit-dialogissa. Dialogissa (ks. kuvio 11) on listattuna tilauskoodit ja niiden yleiset tiedot. Listalta näkee, mitkä koodit asiakas on käyttänyt. Näitä koodeja ei voida enää muokata tai käyttää uudelleen. Käyttäjä voi halutessaan ladata tilauskoodit uudelleen eri tiedostomuodoissa tietokoneelleen.

Order code	Used
NTb49Pqo	No
uDUphTDD	No
uQXGJUiv	No
qEMGSVc9	No
9CQeUpez	No
WPzN2W89	No

Kuvio 11. Tutki tilauskoodeja -dialogi

4.3 Centerien hallinta

”Centerit”-välilehden (ks. kuvio 12) toimintaperiaate on sama kuin pääsivulla.

Vasemmassa listassa on listattuna kaikki Centerit. Kun Center valitaan listalta, sille kuuluvat käyttäjätilit ladataan ja näytetään oikealla Center-käyttäjien listassa.

The screenshot shows the FLA Admin Client interface. At the top, there is a navigation bar with 'Main', 'Centers', and 'Administration' tabs. Below this, there are two main sections: 'Centers' and 'Center users'. Each section has a search bar. The 'Centers' section contains a table with columns 'Name' and 'Edit'. The 'Center users' section contains a table with columns 'Name', 'Role', and 'Edit'. At the bottom, there are buttons for 'Create new Center...', 'Re-send credentials', and 'Create new user...'.

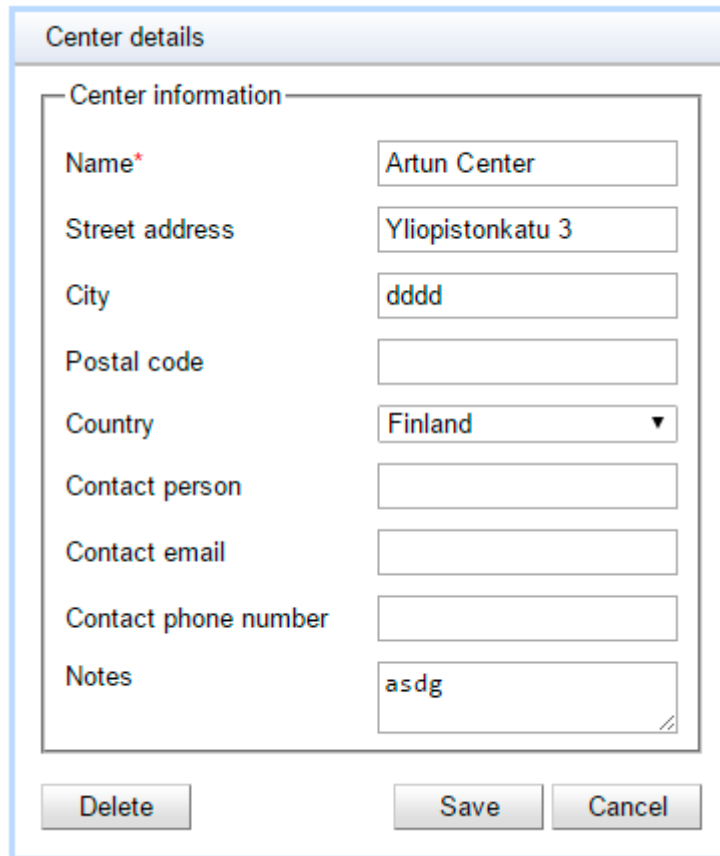
Name	Edit
Artun Center	Edit...
Firstbeat Center	Edit...
Hannun Center1	Edit...
Heinin Center	Edit...
Heinin Center 2	Edit...
Heinin Center 3	Edit...
Ilkan Center	Edit...
New New Center	Edit...
Riikan Center	Edit...
Riikan Center 2	Edit...
SaulinCenter	Edit...
The Center of Marcin	Edit...
The new Center4	Edit...

Name	Role	Edit
dfass, asdf	Logistician	Edit...
fasfasd, Are	Logistician	Edit...
Isop, Arttu	Logistician	Edit...
Testi, Artun	Admin	Edit...

Kuvio 12. Käyttöliittymäkuva hallintajärjestelmän Centerit-välilehdestä

Valittua Centeriä voidaan muokata painamalla ”Muokkaa”-painiketta listassa, jolloin aukeaa Centerin muokkaamiseen ja luomiseen tarkoitettu dialogi (ks. kuvio 13).

Myös Centerin poistaminen toimii myös sitä kautta. Uusi Center voidaan luoda ”Luo uusi Center”-painikkeesta.



Center details

Center information

Name*

Street address

City

Postal code

Country

Contact person

Contact email

Contact phone number

Notes

Delete Save Cancel

Kuvio 13. Centerien muokkaamiseen ja luomiseen käytetty dialogi

Oikealla Center-käyttäjien listassa näytetään valitun Centerin työntekijät, joiden muokkaaminen ja poistaminen toimii dialogin (ks. kuvio 14) kautta samalla tavalla kuin edellä on mainittu. Uusi Center-käyttäjä voidaan luoda "Luo uusi Center-käyttäjä"-painiketta painamalla. Molemmilla listoilla on oma hakukenttä, joilla voidaan hakea kyseisen listan rivejä nimen mukaan.

Center user details

Personal information

First name* Arttu

Last name* Logistikko

Email address* tyystin@mahdoton.net

Username* arttulogi

Role Logistician Admin

Valid until 23.11.2016 X

Delete Save Cancel

Kuvio 14. Center-käyttäjien muokkaamiseen ja luomiseen käytetty dialogi

4.4 Järjestelmänvalvojat

Kolmannessa ”Hallinto”-välilehdessä (ks. kuvio 15) on listattuna kaikki järjestelmänvalvojat. Järjestelmänvalvoja voidaan hakea hakukentän avulla käyttäjänimen perusteella.



Kuvio 15. Käyttöliittymäkuva hallintajärjestelmän Hallinto-välilehdestä

Järjestelmänvalvoja voidaan muokata tai poistaa painamalla "Muokkaa"-painiketta listassa, jolloin järjestelmänvalvojan muokkaamiseen ja lisäämiseen tarkoitettu dialogi (ks. kuvio 16) aukeaa. Uusi järjestelmänvalvoja voidaan luoda "Luo uusi Admin"-painikkeella.

The screenshot shows a dialog box titled "Admin account details". It contains a section for "Admin information" with the following fields:

- Username*: Arttu2
- Email address: arttu.isopahkala@firstbe
- First name: arttu2
- Last name: isopahkala
- Valid until: 16.02.2017 (with a clear button 'x')

At the bottom of the dialog, there are two buttons: "Save" and "Cancel".

Kuvio 16. Järjestelmänvalvojan muokkaamiseen ja luomiseen käytetty dialogi

5 Työn toteutus

5.1 Rajapintakutsut

Hallintajärjestelmä käyttää REST-rajapintaa, jonka avulla haetaan, muokataan, lisätään ja poistetaan tietoa Riak-tietokantaa käyttäen. Rajapintakutsuissa tarvitaan siis myös palvelinpään toteutuksia. Nämä toteutukset eivät kuuluneet opinnäytetyön tekijän tehtäväksi, vaan ne tulivat palvelinpään kehittäjiltä, joiden kanssa tehtiin tiivistä yhteistyötä työprosessin ajan. Rajapintakutsuja toteutettiin sen perusteella, millaista tietoa sovelluksen täytyy hallita ja näyttää.

Sovelluksessa on rajapintakutsuja varten luotuna kaksi tärkeää Java-luokkaa. Toinen on nimeltään AdminServiceRestInterface, joka sisältää polun rajapinnan resurssiin ja pyynnön (request), jota kutsutaan toisen Java-luokan AdminServicen kautta. Tämä Java-luokka sisältää käsittelyn kutsun onnistumiselle ja epäonnistumiselle.

AdminServiceen on ohjelmoitu metodi, jota puolestaan voidaan kutsua esittäjistä käsin. Jos tietoa haetaan, metodi palauttaa onnistuessaan halutun datan.

AdminServiceen ohjelmoituja metodeja käytetään myös datan muokkaamiseen tai poistamiseen. Esimerkiksi asiakastilille kuuluvat käyttäjätilit saadaan kutsumalla esittäjässä metodia "getUsers(customerId, GetUsersHandler)". Parametrina lähetetään asiakkaan id ja kutsun käsittelijä:

```
AdminService.getInstance().getUsers(123, this);
```

Metodi AdminServicessä sisältää aikamääreiden muuntamisen palvelimen aikaan sekä rajapinnan yliajetut kutsut vastausta varten. Koska User-olio sisältää arvoja kellonajoille, muunnetaan aikamääreet palvelimelle UTC (Universal Time Code)-aikaan, jotta eri aikavyöhykkeillä sijaitsevat käyttäjät näkevät päivämäärät oikein.

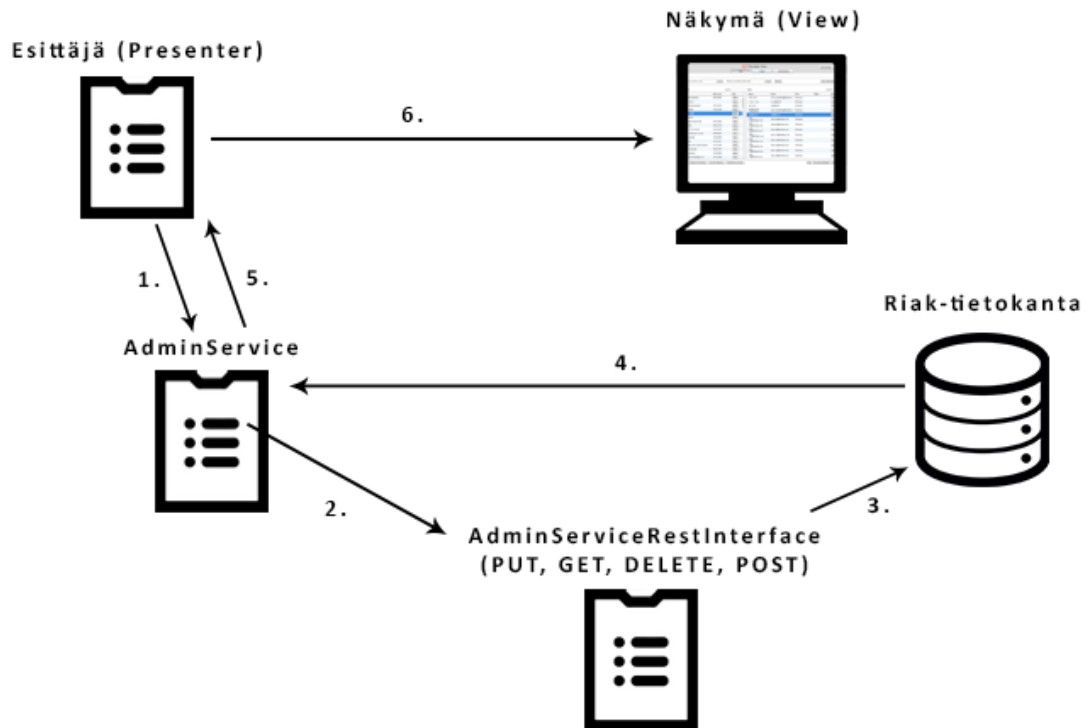
Metodi kutsuu AdminServiceRestInterfacen REST-rajapintakutsua:

```
@GET
@Path("/customers/{customerId}/users")

public void getUsers(@HeaderParam("Authorization") String authHeader, final
@PathParam("customerId") long customerId, MethodCallback<GetUsersResponse> callback);
```

Rajapintakutsun vastaus (response) käsitellään AdminServicessä käsittelijän (GetUsersHandler) avulla, joka kutsuu takaisin joko getUsersSuccess(ArrayList<User>

users) tai getUsersFailure() metodia. Jos kutsu onnistuu, esittäjässä yliajettu "getUsersSuccess"-metodi saa parametrina listan käyttäjätileistä, joita voidaan käsitellä sovelluksessa eri tavoin. Kuviossa 17 on numeroituna oikeaan järjestykseen se, mitä kautta tieto haetaan tietokannasta ja kuinka se kulkeutuu näkymään.



Kuvio 17. Tiedon hakeminen tietokannasta ja sen kulku näkymään

5.2 Käyttöliittymä

Hallintajärjestelmän ulkoasusta pyrittiin tekemään mahdollisimman samanlainen kuin suunnitellut käyttöliittymäkuvat. Tyyliin käytettiin valmiita Firstbeatin tyyliä lisäksi kuitenkin tarvittavia yksityiskohtaisia CSS-tyylejä muutamille käyttöliittymäkomponenteille. Sovelluksessa käytettiin uudehkoa Flex Box -ulkoasumallia, jolla pystytään kätevästi luomaan responsiivisia ulkoasuja. Flex Boxin avulla säiliön komponentit voidaan sijoittaa ja järjestellä säiliöön eri tavoin ilman, että jokaiselle komponentille täytyisi erikseen asettaa etäisyydet. Esimerkiksi tyyli "justify-content: space-around;" järjestee komponentit koko säiliön alueelle tasaisin välimatkoin jättäen jokaiseen väliin yhtä paljon tilaa. Flex Boxin avulla voidaan myös päättää järjestelläänkö komponentit pysty- vai vaakasuunnassa, ja kuinka komponentit sijoittuvat erikokoisilla näyttöpäätteillä.

Käyttöliittymän luontivaiheessa mietittiin mistä on järkevintä aloittaa. Päänäkymän ulkoasu oli paras vaihtoehto, koska oli hyvä saada valmiiksi ensiksi se näkymä, jota kautta avattaisiin dialogeja. Eclipseen ladatun GWT-lisäosan avulla näkymien luonti on kätevää, koska sillä voidaan samaan aikaan luoda automaattisesti sekä XML-tiedosto, että Java-luokka näkymälle, joissa viitataan automaattisesti toisiinsa. Näkymien rakennus aloitetaan poikkeuksetta ensin XML-tiedostosta, johon sijoitetaan komponentit ja niiden tyylimäärittelyt. GWT:n komponentit haetaan UiBinderin avulla Java-tiedostoon, jossa niitä käytetään eri tavoin. Komponenteille asetetaan esimerkiksi lokalisoitu teksti sen tyylistä riippuen ja tarvittaessa tapahtumankäsittelijöitä. Näkymän koodiin kirjoitetaan esittäjää varten tiedon hakijat ja muuttajat, sekä metodit esimerkiksi näkymän piilottamiselle ja tekstin asettamiselle. Näkymä sisältää vain ne metodit, jotka jollaintapaa toteutuessaan näkyvät käyttäjälle tai ottavat vastaan käyttäjän toimia. Käyttöliittymässä käytetyt tekstit ovat nimi-arvopareille tarkoitettussa properties-tiedostossa. Jokaiselle kielelle on luotu oma tiedosto, joita käytetään lokalisoinninkäsittelijän avulla näkymän koodissa.

5.3 Kustomoidut komponentit

Hallintajärjestelmä koostuu pääasiassa valmiista itse tyyllitellyistä perinteisimmistä käyttöliittymäkomponenteista kuten painikkeista, tekstikentistä ja dialogeista. Sovelluksessa on kuitenkin käytetty muutamia kustomoituja komponentteja, joita ovat mm. tietoa sisältävät listat ja päivämäärän valitsinnäkymä.

5.3.1 FbtGrid

GWT sisältää DataGrid Java-luokan, jonka avulla luodaan tietoa sisältäviä listoja. Hallintajärjestelmässä listat ovat suuressa roolissa ja niitä on jokaisella välilehdellä sekä muutamassa dialogissa. Tätä varten sovelluksessa käytettiin DataGrid Java-luokasta laajennettua FbtGrid Java-luokkaa, joka on kustomoitu komponentti. Luokan tarkoitus on helpottaa listan käyttöä ja palvella käyttäjän toimia parhaalla mahdollisella tavalla.

Yksi FbtGridin kustomoinnin syistä oli rivien oikeanlainen valitseminen. Haluttiin, että FbtGridia voidaan käyttää useissa Firstbeatin web-sovelluksissa, joissa on erilaisia

listan valintamahdollisuuksia käyttötarpeen mukaan. Listalle luotiin valinnat yksittäisen rivin valinnalle, valinnalle valintaruutujen kanssa ja valinnalle joko Ctrl- tai Shift-painiketta käyttäen (ks. kuvio 18). Esimerkiksi edellä mainittu yhden rivin valinta tarkoittaa sitä, että listalta voi olla valittuna vain yksi rivi kerrallaan. Valintatyyppi voidaan määrittellä enumeraationa FbtGridin luontivaiheessa.

```
public enum SelectionType {
    SINGLE,
    MULTISELECT_WITH_CHECKBOX,
    MULTISELECT_WITH_CTRL_OR_SHIFT;
}
```

FbtGrid Java-luokka sisältää myös metodin, jolla voidaan luoda kustomoituja soluja listaan. Nämä ovat hyödyllisiä niissä tapauksissa, kun solu sisältää useamman kuin yhden komponentin tai sarakkeelle halutaan erilaisia komponentteja eri riveille. Metodi ottaa vastaan parametrina SafeHtml-tekstin, joka on komponentit sisältävä HTML-kielinen teksti. FbtGridin avulla sarakkeita voidaan luoda vain yhtä metodia käyttäen, johon lähetetään parametrina tarvittavat arvot. Näin listan luonti onnistuu helpommin. Java-luokassa on lisäksi muita pienempiä ominaisuuksia, kuten rivien järjestelyn ja rivivalinnan muuttumisen tapahtumankäsittelijät.

Users Search

Name	Email	▲ Role	Office	Edit
1234, 123	arttu.isopahkala@firstbeat.co	Customer Admin	Toimisto 2	Edit...
asd, Artturi	arttu.isopahkala@firstbeat.co	Customer Admin		Edit...
Isopahkala, Arttu	arttu.isopahkala@firstbeat.co	Customer Admin		Edit...
arttu.isopahkala@firstbea	arttu.isopahkala@firstbeat.co	Pro		Edit...
asd, asd	arttu.isopahkala@firstbeat.co	Pro		Edit...
PRO, TEST	tyystin@mahdoton.net	Pro		Edit...
<zxcv, <xcz	xcvz@asd.fi	End-user	Toimisto 1	Edit...
aasd@dg.dgi	aasd@dg.dgi	End-user		Edit...
ads, asd	asd@das.fi	End-user		Edit...
adsdas1234, asdasd1234	arttu.isopahkala@firstbeat.co	End-user		Edit...
afs@asd.sv	afs@asd.sv	End-user		Edit...
arttu-b2@firstbeat.com	arttu-b2@firstbeat.com	End-user		Edit...
arttu-b3@firstbeat.com	arttu-b3@firstbeat.com	End-user		Edit...
arttu-b4@firstbeat.com	arttu-b4@firstbeat.com	End-user		Edit...

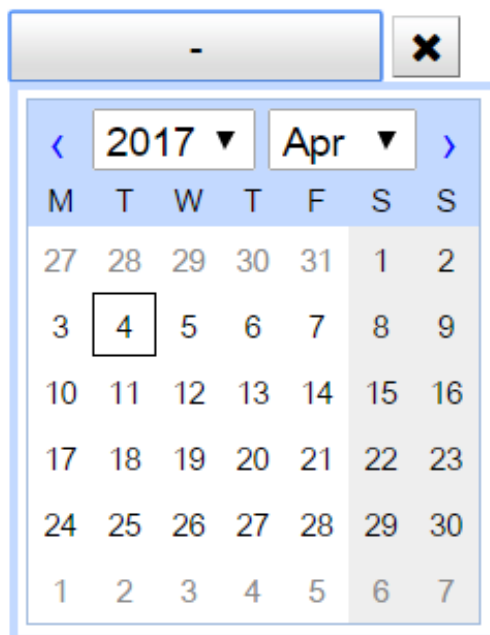
Kuvio 18. FbtGrid luotu käyttäen "MULTISELECT_WITH_CTRL_OR_SHIFT"-enumeraatiota

5.3.2 FbtDateBox

Dialogeissa tarvittiin päivämäärän valitsemista esimerkiksi asiakastilin voimassaololle ja tilastojen rajaamiselle tietylle aikavälille. Tätä varten käytettiin GWT:n staattisesta DateBox-kalenterikomponentista laajennettua FbtDateBox Java-luokkaa. Luokka sisältää muutaman ominaisuuden, joiden avulla kalenterinäkömä (ks. kuvio 19) voidaan alustaa halutunlaiseksi:

- Kalenterinäkömän viereen voidaan asettaa rasti, jolla tyhjennetään päivämäärä.
- Kalenterin päivämäärä voidaan alustaa joko tyhjäksi tai johonkin tiettyyn päivämäärään.
- Kalenterin päivämäärä voidaan muotoilla numeeriseen tai osittain sanalliseen muotoon

FbtDateBox sisältää lisäksi tapahtumankäsittelijöitä, joihin voidaan tarttua esittäjistä. Tapahtumankäsittelijä on liitetty kalenterin aukeamiseen, päivämäärän vaihtumiseen kalenterinäkömässä ja painikkeeseen, joka tyhjentää päivämäärän.



Kuvio 19. FbtDateBox-kalenteri, jossa rasti päivämäärän tyhjentämistä varten

5.4 Dialogit

Hallintajärjestelmän käyttöliittymätason suurin toiminnallinen aihealue on dialogit. Lähes kaikki toiminnot hoidetaan dialogin kautta. Tästä kertoo se, että melkein jokaisesta painikkeesta aukeaa jonkinlainen dialogi. Seuraavassa esimerkki kertoo kuinka Center-käyttäjän lisäämiseen ja muokkaamiseen tarkoitettu dialogi luotiin käyttäen GWT:n UiBindausta ja MVP-arkkitehtuuria. Eclipsen GWT-lisäosan avulla dialogille luotu näkymä sisältää aina kaksi osaa. Toinen niistä on XML-puurakenne, johon sijoitellaan komponentit sekä niiden tyylit oikeaan järjestykseen.

```
<!DOCTYPE ui:UiBinder SYSTEM "http://dl.google.com/gwt/DTD/xhtml.ent">
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.uibinder" xmlns:g="urn:import:com.google.gwt.user.client.ui" xmlns:v="urn:import:fi.firstbeat.gwt.client.view.admin">
<g:HTMLPanel addStyleNames="width350px">
<div class="dialogHeader">
<g:Label ui:field="centerUserWelcome" />
</div>
<div class="displayFlex flexDirectionColumn padding5px">
<g:CaptionPanel ui:field="centerUserInfoTitle" addStyleNames="margin5px">
...

```

Toinen osa on Java-luokka, joka käyttää UiBinderiä hakeakseen komponentit XML-tiedostosta.

```
public class AdminClientCenterUserDialog extends Composite {
private static AdminClientCenterUserDialogUiBinder uiBinder =
GWT.create(AdminClientCenterUserDialogUiBinder.class);
interface AdminClientCenterUserDialogUiBinder extends UiBinder<Widget,
AdminClientCenterUserDialog> {}
@UiField Label centerUserWelcome;
@UiField CaptionPanel centerUserInfoTitle;
...

```

Jokainen näkymä, joka sisältää staattista tekstiä, sisältää "localize"-metodin. Metodi hakee lokalisoinninkäsittelijän (LocalizationHandler) avulla resursseista tekstit käyttöliittymäkomponentteihin. Esimerkiksi "tallenna"-painikkeeseen teksti haetaan seuraavasti:

```
AdminClientLocalizationHandler LH = AdminClientLocalizationHandler.get-
Instance();

saveButton.setText(LH.getText("Common.Button.Save"));
```

Painikkeet ja muut käyttöliittymäkomponentit käsittelevät käyttäjän toimintoja, joten niihin täytyy asettaa erilaisia tapahtumankäsittelijöitä.

```
@UiHandler("saveButton")
void onSaveButtonClick(ClickEvent event) {
    presenter.onSaveButtonClick();
}
```

Esimerkissä käytetty tapahtumankäsittelijä lähettää rajapinnan (interface) kautta esittäjälle tiedon, että käyttäjä on painanut painiketta. Jokainen näkymä tarvitsee rajapinnan tai tapahtumanlaukaisun (event) suorittaakseen käyttäjän toiminnon. Rajapinta on yleensä erillisessä tiedostossa rakenteellisista syistä. Rajapinnan metodeita kutsutaan näkymän tapahtumankäsittelijöistä käsin.

```
public interface AdminClientCenterUserDialogPresenterInterface {
    public void onSaveButtonClick();
    public void onCancelButtonClick();
    public void onDeleteButtonClick();
}
```

Näkymä liitetään rajapinnan avulla esittäjään kutsumalla esittäjästä näkymän "setPresenter"-metodia, johon lähetetään parametrina rajapinta. Metodien tarkoitus on kertoa, että esittäjä haluaa käyttää kyseistä näkymää ja käsitellä sen toimia.

```
public void setPresenter(AdminClientCenterUserDialogPresenterInterface pre-
senter) {
    this.presenter = presenter;
}
```

Esittäjässä sen käyttämä näkymä alustetaan tiedoston alussa. "Show"-metodissa asetetaan rajapinta näkymään ja lisätään parametrina lähetetty näkymä säiliöön (tässä tapauksessa ModalDialog-komponenttiin):

```
private AdminClientCenterUserDialog view;
public void show(HasWidgets container) {
    view.setPresenter(this);
```



```

...
container.clear();
container.add(view);
}

```

Esittäjän täytyy implementoida ja yliajaa rajapinnan metodit. Metodien tarkoitus on muuttaa mallia tai näkymää. Tässä tapauksessa tallentaa käyttäjän.

```

public class AdminClientCenterUserDialogPresenter implements AdminClientCenterUserDialogPresenterInterface
...
@Override
public void onSaveButtonClick() {
AdminService.getInstance.saveUser(user, this);
}

```

Samaa dialogia voidaan käyttää erilaisissa käyttötarkoituksissa, muuntamalla sen näkymää halutunlaiseksi. Näin säästytään ylimääräiseltä työltä ja ohjelmoinnilta. Dialogin esittäjään ohjelmoidaan metodit, joita kutsutaan eri tapauksissa. Dialogille voidaan asettaa esimerkiksi enumeraation muodossa näkymän tila muokkausta varten tai pelkkää katselua varten. Jos käyttäjälle annetaan oikeus vain katsella dialogia tekemättä siihen muutoksia, kutsutaan esittäjän metodia katselu-tilalle, joka estää näkymän muokkaamisen.

6 Tulokset

6.1 Tulos tavoitteisiin nähden

Suunnitelmaan nähden tavoitteet täyttyivät lähes täydellisesti. Hallintajärjestelmään saatiin sisällytettyä kaikki tärkeimmät ominaisuudet mitkä alun perin suunniteltiin. Lopputuloksena hallintajärjestelmää pystyttiin käyttämään verkkoselaimen avulla ja vanhaa järjestelmää ei enää tarvittu. Ennen uuden hallintajärjestelmän julkaisua vanhaa järjestelmää päivitettiin aina sen mukaan millaisia uusia ominaisuuksia tarvittiin. Suuremmat muutokset päivitettiin ainoastaan uuteen järjestelmään, koska tiedettiin, että vanhan järjestelmän käyttö tullaan lopettamaan. Uuden hallintajärjestelmän valmistuttua vanha hallintajärjestelmä poistettiin käytöstä.

Vaatimusmäärittely muuttui jonkin verran työn toteutuksen aikana, joka vaati lisätyötä suunnitteluun ja ohjelmointiin. Joitain ominaisuuksia toteutettiin niin myöhään, että ne jätettiin tästä opinnäytetyöstä pois. Hallintajärjestelmään esimerkiksi luotiin uusi välilehti uutta käyttäjäryhmää varten. Nämä käyttäjät olivat Firstbeatin työntekijöitä, jotka saivat oikeuden käyttää uutena tullutta Analyysikeskusta, jota kautta asiakkaiden mittauksia analysoidaan ja niistä luodaan raportteja. Välilehti toimi samalla periaatteella kuin hallinto-välilehti ja käyttäjien tiedot olivat suunnilleen samat kuin järjestelmänvalvojilla.

Hallintajärjestelmän käyttöliittymä oli useiden keskusteluiden jälkeen syntynyt toteutus, joka tyydytti mahdollisimman monta järjestelmän käyttäjää. Käyttäjille annetaan enemmän tietoa järjestelmän toiminnoista kuin ennen. Pienet informatiiviset dialogit kertovat toiminnon jälkeen tai sitä tehdessä, mihin toiminto tulee vaikuttamaan tai onnistuiko se. Hallintajärjestelmän ulkoasusta saatiin lähes samannäköinen, mitä käyttöliittymäkuviin suunniteltiin. Tarkoitus ei ollut kuitenkaan tehdä käyttöliittymästä pikselin tarkkuudella samanlaisia kuin kuvat, koska suunnitelmat eivät olleet niin tarkasti tehtyjä. Aikaa olisi muuten kulunut turhaan suunnitteluun, joka olisi jäänyt toteutukselta pois. Ohjelmoidessa kuitenkin aseteltiin etäisyydet ja komponenttien koot parhaaksi katsotulla tavalla ja suunnitelmista ei tarvinnut pahemmin poiketa.

Koska hallintajärjestelmä käännettiin sekä suomeksi että englanniksi, syntyi komponentteihinkin pieniä eroavaisuuksia tekstien pituuden takia. Suomeksi sanat ovat usein pidempiä kuin englanniksi, joten jotkin painikkeet ja säiliöt ovat normaalia leveämpiä kuin suunniteltiin. Tätä ongelmaa ratkottiin lyhentämällä joitain suomenkielisiä sanoja ja asettamalla etäisyyksiä hieman eri tavoin, jotta lopputulos toimisi hyvin molemmilla kielillä.

6.2 Vertailu vanhaan hallintajärjestelmään

Uudesta hallintajärjestelmästä jätettiin pois ”Customizing” ja ”Operation Statistics” välilehdet, koska niitä käytettiin erittäin harvoin tai ei ollenkaan. Suurin rakenteellinen muutos vanhaan hallintajärjestelmään nähden on tiedon esittäminen. Vanhassa järjestelmässä asiakas- ja käyttäjätilit näkyivät eri välilehdissä, kun taas uudessa järjestelmässä samalla sivulla. Kun vanhassa hallintajärjestelmässä asiakas-

tai käyttäjätili valitaan listalta, tulevat sen tiedot listan vieressä oleviin kenttiin. Näin tiliä voitiin muokata nopeasti. Tämä kuitenkin teki käyttöliittymästä sekavan, koska aina ei tiedetty mitä tiliä muokataan, mitä tietoa tallennetaan ja mihin tallennetaan. Tästä syystä tilien muokkaaminen siirrettiin dialogeihin, jolloin käyttö on selkeämpää. Asiakas- ja käyttäjätilit siirrettiin samalle sivulle, jotta asiakkaan käyttäjätileihin päästään heti käsiksi.

Asiakastilien etsimisessä käytetyt suodattimet tilin voimassaolon ja managerin mukaan jätettiin pois. Asiakastilin tietojen muokkaamisesta jätettiin tarpeettomuuden takia pois seuraavat valinnat tai kentät:

- Verkkotunnukset
- Käytä esikyselylomaketta
- Käytä palautekyselyä
- Kysy tavoitteet
- Lähetä automaattisesti linkit hyvinvointianalyseissä
- Salli linkkien lähetykset hyvinvointianalyseissä
- Ajanvaraus ja siihen liittyvät päivämäärärajoitukset
- Kysy lupa tietojen luovuttamiseen
- Estä raporttien luonti
- Pääsy raporttieditoriin

Myös Firstbeat Center valintaruutu jätettiin pois, koska valinta suunniteltiin järkevämmäksi. Center valitaan asiakastilille alavetovalikon kautta, josta joko valitaan jokin Center tai ei valita mitään Centeriä. Dialogista, jossa luodaan laitetilaukskoodeja, jätettiin pois sähköpostikutsuihin liittyvät toiminnot, koska niitä käytettiin niin harvoin. Tilalle tuli ominaisuus, jolla voidaan luoda tilauskoodeista Excel-tiedosto. Vanhan hallintajärjestelmän asiakastilit-näkymässä ollut painike, joka näytti asiakastilille varatut mittalaitteet, jätettiin myös pois.

Käyttäjätilin tiedoista poistettiin voimassaoloajan sekä Uploader-roolin valinta. Tietoihin lisättiin toimiston valinta sekä tiedot käyttäjätilin luojasta, käyttäjätilin luomisajasta sekä viimeisestä kirjautumisajasta. Käyttäjätili-listaan laitettiin esille rooli ja toimisto, joiden mukaan tilejä on helppo järjestellä listassa painamalla sarakkeen otsikkoa. Center-käyttäjiltä jätettiin kielen ja aikavyöhykkeen valinta pois. Hallinto-näkymään tehtiin myös muutamia muutoksia. Järjestelmänvalvoja ei voida uudessa hallintajärjestelmässä poistaa. Oman tilin asetukset löytyy erillisen ”Oma

tili"-painikkeen takaa. Järjestelmänvalvojalta jätettiin pois puhelinnumeron asettaminen sekä viimeisimmän kirjautumisajan näyttäminen.

Kokonaisuudessaan pois jätettiin paljon eri asioita, joita oli vanhassa hallintajärjestelmässä. Osa poisjätetyistä toiminnoista herätti keskustelua ja voi olla, että tulevaisuudessa uuteen hallintajärjestelmään tullaan lisäämään vielä jotain vanhan järjestelmän ominaisuuksista. Toistaiseksi kuitenkin lisättiin kaikista tärkeimmät ominaisuudet. Myös toiveiden ja ehdotusten perusteella uuteen hallintajärjestelmään luotiin uusia ominaisuuksia, jotka eivät välttämättä ennen olleet niin tarpeellisia kuin nyt.

6.3 Työn hyödyt yritykselle

Yrityksellä on monia web-sovelluksia sekä asiakkaiden, että työntekijöiden käytössä. Hallintajärjestelmä oli ainoita yhä käytössä olevia sovelluksia, joita ei vielä ollut päivitetty web-ympäristöön. Tästä syystä oli tärkeää, että sekin saatiin päivitettyä nykyaikaisempaan muottiin GWT:tä käyttäen. Tästä edes järjestelmänvalvojat voivat kirjautua hallintajärjestelmään millä tahansa tietokoneella vain verkkoselainta käyttäen, joten sen käyttö on nopeampaa.

Tilien ja centerien hallinta on nyt selkeämpää ja helpompaa vanhempaan hallintajärjestelmään verrattuna, koska turhat ominaisuudet karsittiin pois. Vanhasta hallintajärjestelmästä oli tullut palautetta, että joitain tietoja ei uskaltanut muokata, koska käyttöliittymä oli jokseenkin sekava. Uuden hallintajärjestelmän kohdalla jokainen käyttäjä sai antaa palautetta ja vaikuttaa tarpeellisiin ominaisuuksiin. Tämä auttoi räätälöimään käyttöliittymästä ja toiminnoista sellaisen, johon jokainen voisi olla tyytyväinen.

7 Pohdinta

Opinnäytetyön toteutus oli aloittelevalla ohjelmoijalla opettavainen kokemus. Ehkä suurin hyöty jäi MVP-arkkitehtuurimallin käyttämisestä. Se toi paljon järjestystä koodien sijoitteluun ja helpotti työn tekoa. Kun laajahkon hallintajärjestelmän eri osa-alueita ohjelmoitiin, tiedettiin aina mistä tiedostosta löytää halutun toiminnallisuuden tai käyttöliittymän koodit. Oli myös erittäin kätevää käyttää

käsittelijöitä (handler) kutsumaan takaisin rajapintakutsujen tuomaa tietoa. Ne voitiin asettaa niihin tiedostoihin järjestelmässä, joissa tiedettiin, että tietoa tarvittiin käsitellä tai hakea. MVP-arkkitehtuurimallin käyttö tai soveltaminen tulee olemaan erittäin käytännöllistä myös tulevien web- ja mobiilisovellusten teossa.

Hyödyllistä työssä oli oppia paljon uusia tekniikoita, myös palvelinpään asioita. GWT:llä luotu sovellus täytyi "buildata" eli rakentaa palvelimelle, missä käytin hyödyksi skriptejä. Skripti oli BAT-tiedostossa, joka kokosi luodun web-sovelluksen Ant-tiedoston avulla yhtenäiseksi paketiksi ja siirsi palvelimelle web-ympäristöön. Vaikka GWT ei olekaan suosituin Java-kehitystyökalu tänä päivänä, voisin hyvin käyttää sitä lisääkin, koska sen avulla oli helppo ohjelmoida monimutkainen web-sivusto Java-ohjelmointikielellä. Varsinkin kun tarvitaan laajaa tietokantaa ja tiedon muokkausta, GWT ja Java-kieli nousivat esiin.

Aikaa oli suunniteltu käytettäväksi sopivasti ja hallintajärjestelmä valmistui ajallaan. Tätä edistivät myös firman muut työntekijät, joilta sai neuvoja ohjelmointiin ja ominaisuuksien suunnitteluun. Myös palvelinpään ohjelmoijien kanssa välillä hyvinkin tiivis yhteistyö sujui ilman ristiriitoja. Opinnäytetyön loppuvaiheessa hallintajärjestelmään lisättiin myös lisää ominaisuuksia, kuten uusi välilehti Analyysikeskuksen käyttäjille ja lisää tilikohtaisia asetuksia.

Kokonaisuudessaan työ sisälsi monta osa-aluetta. Suunnitteluvaiheessa tarvittiin kuulla ihmisten mielipiteitä ja ehdotuksia ominaisuuksista. Mitä enemmän työntekijöitä yrityksessä on, sitä enemmän on myös mielipiteitä ja erilaisia näkökulmia. Tämä vei myös osan ajasta, koska suunnittelu oli tehtävä hyvin. Käyttöliittymäkuvat ja käyttötapauskuvaukset olivat lopulta hyödyllisiä, jotta pysyi itse kartalla siitä miltä järjestelmän tulisi näyttää ja toimia. Tässäkin tapauksessa päti sanonta: "Hyvin suunniteltu on puoliksi tehty". Näin työn toteutus kävi nopeasti, kun myös palvelinpään ohjelmoijat tiesivät mitä haluttiin.

Lähteet

About the Eclipse Foundation. 2017. Eclipsen verkkosivusto. Viitattu 13.2.2017.

<https://www.eclipse.org/org/>

Lehtinen, A. 2011. Malli-näkymä-arkkitehtuurin soveltaminen WPF-teknologialla.

Diplomityö. Viitattu 5.2.2017.

<https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/20983/lehtinen.pdf?sequence=3>

Maple, S. 2016. Java Tools and Technologies Landscape Report 2016. Viitattu

28.3.2017. <https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/>

Overview. 2017. GWT. Viitattu 5.2.2017. <http://www.gwtproject.org/overview.html>

Photoshop Reigns Supreme: How the Software Has Maintained Market Dominance.

2017. Verkkosivusto. Viitattu 30.3.2017.

<https://www.business.com/articles/photoshop-reigns-supreme-how-the-software-has-maintained-market-dominance/>

Ramsdale, C. 2010. Building MVP apps: MVP Part I. GWT:n verkkosivusto. Viitattu

14.2.2017. <http://www.gwtproject.org/articles/mvp-architecture.html>

The Future of GWT Report. 2015. Vaadin verkkosivusto. Viitattu 28.3.2017.

https://vaadin.com/documents/10187/4238532/GWT_report_2015.pdf/066936d9-8fcf-40bb-a58f-616a3d3f8203

Valtakunnallinen yrittäjäpalkinto Firstbeat Oy:lle. 2016. Verkkosivusto. Viitattu

22.3.2017. <https://www.yrittajat.fi/keski-suomen-yrittajat/a/uutiset/542674-valtakunnallinen-yrittajapalkinto-firstbeat-oylle>

Yritys. 2017. Firstbeatin verkkosivusto. Viitattu 26.1.2017.

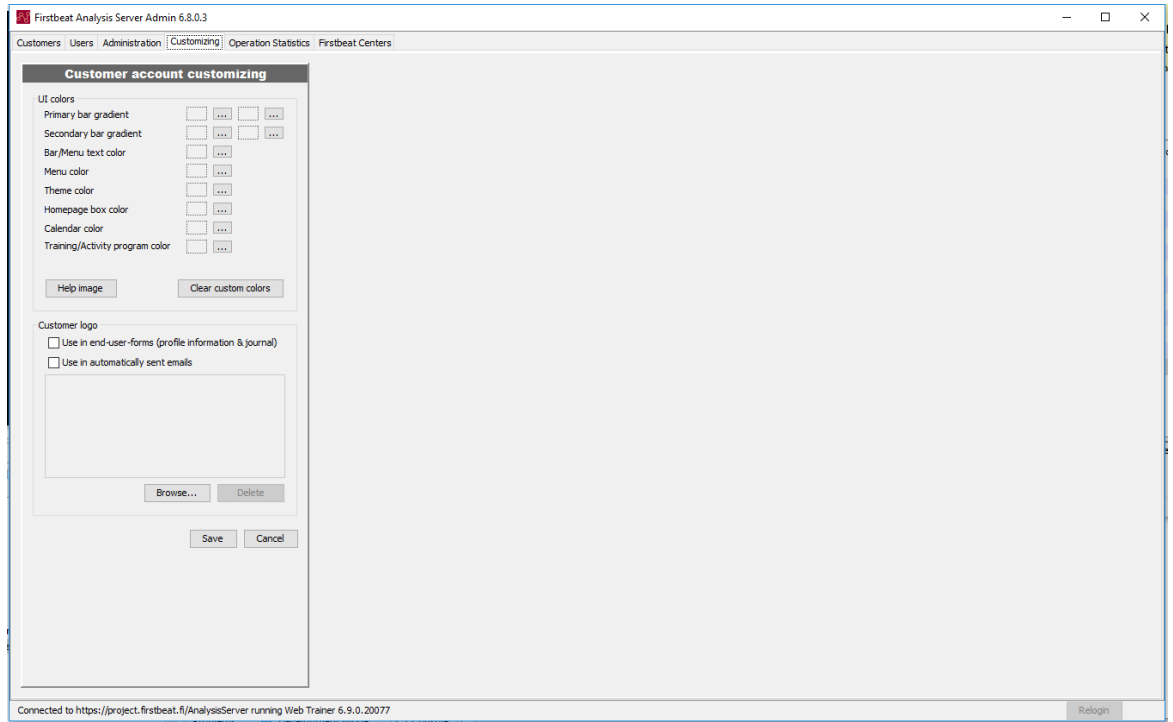
<https://www.firstbeat.com/fi/yritys/>

Ziomacki, P. 2016. Model-View-Presenter Architecture in Android Applications.

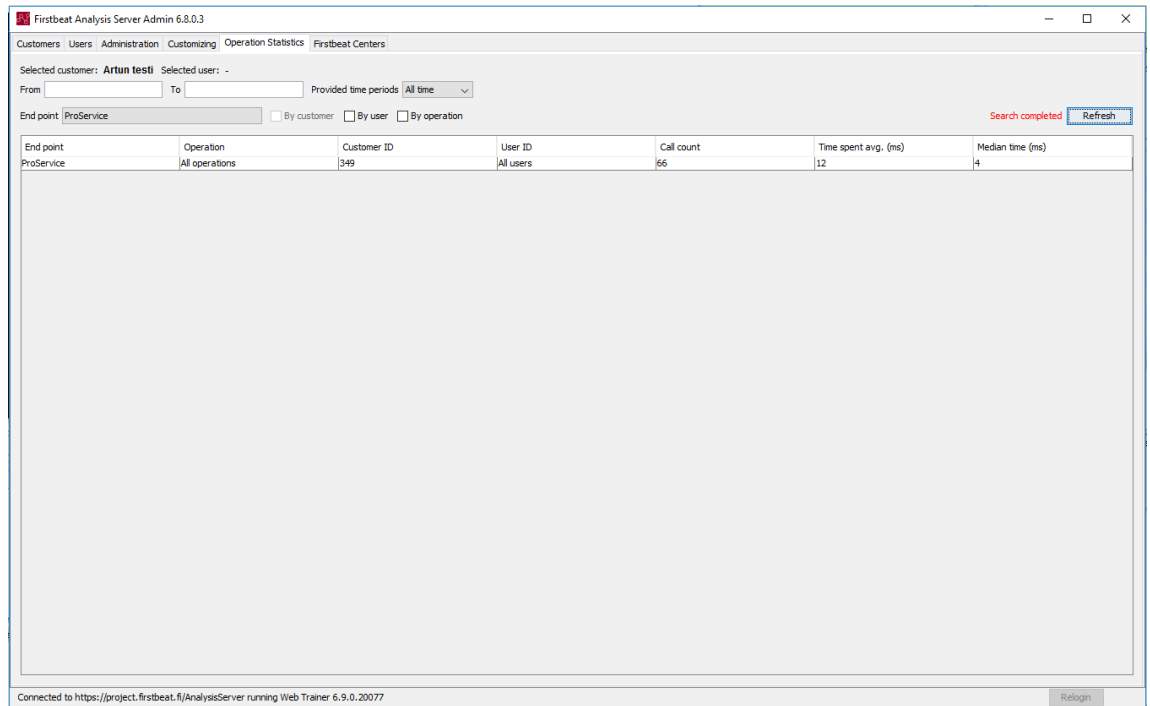
Verkkosivusto. Viitattu 14.2.2017. <http://macoscope.com/blog/model-view-presenter-architecture-in-android-applications/>

Liitteet

Liite 1. "Customizing"-välilehti vanhassa hallintajärjestelmässä



Liite 2. "Operation Statistics"-välilehti vanhassa hallintajärjestelmässä



Firstbeat Analysis Server Admin 6.8.0.3

Customers Users Administration Customizing Operation Statistics Firstbeat Centers

Selected customer: **Artun testi** Selected user: -

From: [] To: [] Provided time periods: All time [v]

End point: ProService By customer By user By operation Search completed Refresh

End point	Operation	Customer ID	User ID	Call count	Time spent avg. (ms)	Median time (ms)
ProService	All operations	349	All users	66	12	4

Connected to https://project.firstbeat.fi/AnalysisServer running Web Trainer 6.9.0.20077 Relogin

Liite 3. Käyttötapauskuvaukset

Main view (*main-view.png*):

General:

1. "Search account by user" -tool searches users with a keyword from all customer accounts. Found users are shown in the users list and accounts where they belong are listed in the accounts list. If this search is used it can cause a situation where customer account is not selected, but users are visible. If customer account is not selected and user tries to edit user account, show error dialog "User cannot be edited because customer is not selected." If search field is empty and "Search" pressed, hide users and show all customers.
2. "Search account by order code" -tool searches customer account by order code. If keyword equals order code, there is only one customer so open the one's "Created order codes" -dialog
3. "Sales information overview" -dialog(*sales-information-overview.png*):
 - When "Sales information overview" button is clicked, open this dialog.
 - When "Export" is clicked, create excel of selected year and close dialog.

Accounts grid:

- Three columns: Name(sortable), Valid until(sortable) and Edit(not sortable)
- In Name column show customer account name. If customer account has notes, show paper icon next to name. If even one user in customer account uses Easy billing mobil, show (Easy) text next to name. If user moves the mouse over a paper icon, show account's notes in a small popup dialog.
- When user clicks a row, show selected customer's user accounts in the users list.
- If user clicks "Edit..." button, open customer account's details in an account dialog.
- If rows are not selected, disable "Account statistics...", "History...", "Report for invoicing", "Move", "Re-send credentials" and "Create new user..." buttons.
- If row is selected, enable "Create new user..." and all buttons below accounts grid.
- Target super search to account name. User can search also with keyword "Easy" which shows accounts with easy billing mode users.

Account dialog (*account-dialog.png*):

This dialog is used for create or edit customer accounts.

- Account name and Max professional users are required fields.
- Account name is unique and may not exist.
- Show all valid system admins in a "manager" list.
- In a "default professional" list show customer's professional and customer admin users.
- Validate "emails for automatic reporting": if emails are incorrect or not separated with space, show error message below the textbox.
- If "Manage..." button is clicked, open "Offices" -dialog
- Show all centers in "center" list. If user selects "<none>", enable report credits field and "add" button and disable center credits field, "add" button and "codes" button. If user selects any center, enable center credit field, "add" and "codes" buttons and

disable report credits field and "add" button. If center credits are added and user selects <none>, show error dialog and remove added center credits. Do same vice versa if report credits are added and center is selected.

- Max professional users value can't be less than pro and customer admin count in a customer account. Value must be a number.
- User must provide values for both valid from and valid until fields or not set them at all. Valid until date can't be earlier than valid from date. Both dates must be in 2000 century. Date selection will be cleared when cross is clicked.
- "Billing customer id", "report credits" and "center credits" fields value must be a number(or empty).
- When "Add" button clicked, add or subtract amount of credits depending if value is positive or negative number.
- If some fields has invalid values, show error message below it. Save account information when "Save" is clicked.
- If dialog has changes and "Cancel" is clicked, show confirmation dialog if user really wants to leave without saving.
- "Delete" is clicked -> delete customer account and its user accounts.

Account dialog opened from "Create new account...":

- Set country to Finland and check "ectopic beats.." checkbox as default.
- If user selects some other country, uncheck "ectopic beats.." checkbox.
- Disable "codes" button. Only existing account can create codes.
- Disable "Delete" button.
- Clear "Valid from" and "Valid until" boxes.

"Offices" -dialog (*offices-dialog.png*):

- List all customer account offices if there is any
- Name column is sortable
- Open "Edit offices" -dialog if user clicks "Edit..." button on a row or "Create new office..." button

"Edit offices" -dialog (*offices-edit-dialog.png*):

- If dialog opens from "Edit..." button, enable "Delete" button and show office's name in a text field
- If dialog opens from "Create new office..." button, disable "Delete" button
- Save office and close dialog if "Save" button is pressed

"Codes" -dialog (*codes.png*):

- Show created order codes' details in this list.
- Sortable columns are "Created", "Expires", "Language" and "Used/Total". Used/Total sorts rows like how many codes can still be used.
- Open "Created order codes" dialog(see below) when user clicks "open" button.
- Open "Edit order codes"(see below) dialog when user clicks "Create order codes" button.

"Created order codes" -dialog (*kuva-puuttuu*):

- Show details on left. Show order codes in a list.
- If code has been used, disable row. If all codes have been used, disable "Edit" button.
- Disable "Create PDF" and "Create Excel" buttons if no codes are selected on the list.
- If there is one or more unused codes selected, enable "Create PDF" and "Create Excel" buttons.
- When "Create PDF" is clicked, export PDF of selected codes.
- When "Create Excel" is clicked, export Excel of selected codes. Excel contains expiration date, group name or specialist name and order codes.
- Open "Edit order codes" dialog when "Edit" button is clicked.

"Edit order codes" -dialog (*edit-order-codes.png*):

This dialog is used for edit or create order codes.

Create codes:

- If user comes to dialog via "Create order codes", show "Number of codes" and "Create only Excel" fields.
- As default expires date is one year ahead.
- Languages are finnish, english and swedish.
- "Professional" list includes professionals and customer admins of customer account.
- "Group" list includes groups of customer account.
- If number of codes is more than center credits on customer account, show confirmation dialog does user really want to create them.
- If "Create only Excel" checkbox is checked, export only Excel(not PDF)

Edit codes:

- If user comes to dialog via "Edit" button in "Created order codes dialog", hide "Number of codes" and "Create only Excel" fields.

Validation:

- Number of codes can't be empty and it must be a number.
- If professional radio button is checked, list selection can't be empty.
- If group radio button is checked, list selection can't be empty.
- Expires date can't be in the past.

"Accounts statistics" -dialog (*account-statistics.png*):

- Show statistics of selected customer account.
- If time period combo box selection is changed, set calendar dates as selected period and refresh statistics.
- If "Refresh statistics" button is clicked, filter statistics by provided calendar dates and set combo box to "<Select>" value.

"Report for invoicing" -dialog (*report-for-invoicing.png*):

- Calendar from date is as default one year in the past.
- If "Create report..." is clicked, export PDF report in provided dates and close dialog.
- Show error if "From" date is later than "To" date.

"History" -dialog (*history.png*):

- All columns are sortable.
- Show all time log history of customer account.

“Refresh” button click:

- Clear user grid and search terms and refresh accounts grid.
- Set buttons to initial state(as in the beginning).

Users grid:

- Five columns: Name(sortable), Email(sortable), Role(sortable as Customer Admin > Pro > End-user), Office(sortable), Edit(not sortable)
- If rows are not selected, disable "Move" and "Re-send credentials" buttons.
- If one row is selected, enable "Move" and "Re-send credentials" buttons.
- If more than one row is selected, enable "Move" button and disable "Re-send credentials" button.
- If user clicks "Edit..." button, open user account's details in user dialog.
- Target super search to user's name and email

"Move" -dialog (*move.png*):

- List all customer accounts to list.
- When "OK" is clicked, move selected user accounts to selected customer account, show selected customer account on the grid and highlight moved user account(s).

“Re-send credentials” button click:

- Send new credentials to user's email.
- If user account has no email or username, show error dialog and don't send anything.

User -dialog (*user-dialog.png*):

- When user clicks "Edit..." on the grid or "Create new user..." show this dialog.
- If dialog is opened from "Create new user..." button, disable "Delete" and "Export measurements" buttons.
- If dialog is opened from "Edit..." button, enable them.
- E-mail address is required field. Check if email is valid.
- If user role is greater than end-user, show "Office" selection. If user is end-user, hide it.
- If "Manual person id" checkbox is checked, the field next to it is required.
- If "Role" selection is "End-user", disable "Username" and "New password" fields and "Easy billing model" checkbox. If "Super Pro", "Customer Admin" or "Pro" is selected, enable "Username" and "New password" fields and "Easy billing model" checkbox. Then username is required field. If password is empty, server generates new one and send credentials to user. If new "Super Pro", "Customer Admin" or "Pro" user is tried to save and customer account doesn't have any "free professional users", show error and don't save user.
- If "Save" is clicked and user account has "Easy billing model" checked and customer account is not "Billable", show info dialog that "Customer is now billable..." and set customer account's "Billable" checkbox checked.

- Delete user account and refresh user grid if "Delete" button is clicked.
- Export .fbc file when "Export measurements" button is clicked. Show info dialog, if user has no measurements to export.

Centers view (*centers.png*)

Centers grid:

- Three columns: Name(sortable), Valid(sortable) and Edit(not sortable)
- In "Name" column show center's name. If center has notes, show paper icon next to name. If user moves the mouse over a paper icon, show center's notes in a small popup dialog.
- If rows are not selected, disable "Re-send credentials" and "Create new user..." buttons.
- When user clicks a row, show selected center's user accounts in the "Center users" list, enable "Create new user..." button and clear Center users search box if it contains text.
- In "Valid" column show if center is disabled or enabled.
- If "Edit..." button is clicked, open center's details in a center dialog.
- Target "Centers" search box to Name column.

Center dialog (*center-dialog.png*):

- If user opens this dialog from "Create new Center...", disable "Delete" button.
- Set Finland as default country.
- If user opens dialog from "Edit.." button, enable "Delete" button.
- Delete Center when "Delete" is clicked.
- Validation when "Save" button is clicked: "Name" field is required and Center's name cannot exist.

Center users grid:

- Three columns: Name(sortable), Role(sortable) and Edit(not sortable)
- If there are no rows selected, disable "Re-send credentials" button.
- If one row is selected, enable "Re-send credentials" button.
- If "Re-send credentials" button is clicked, send new credentials to selected center user's email.
- If "Edit..." button is clicked, show selected user's details in a Center user dialog.
- Target "Center users" search box to Name column.

Center user dialog (*center-user-dialog.png*):

- If user opens this dialog from "Create new user...", disable "Delete" button, clear "Valid until" box and set "Logistician" radiobutton checked.
- Required fields are "First name", "Last name", "Email address" and "Username".
- Check that email address is valid.
- Check that username is valid and doesn't exist.
- If "Valid until" is set, check that date is in this century.
- Delete center user if "Delete" is clicked.

Administration view (*administration.png*)

Admins grid:

- Three columns: Username(sortable), Valid until(sortable) and Edit(not sortable)
- Open Admin dialog if "Edit..." button is clicked.
- Target search box to Username column.

Admin dialog (*admin-dialog.png*):

- If user opens this dialog from "Create new admin...", disable "Delete" button and clear "Valid until" box.
- Username is required. Check that username is valid and not exists.
- Check that "Valid until" date is in this century.
- If user opens dialog from "Edit..." button, disable Username field(it can't be changed) and enable "Delete" button.

My account (*my-account-dialog.png*):

When user clicks "My account" on the top right corner, this dialog opens.

- Disable Username field.
- Check that Email address is valid.
- Check that valid until is not in the past and it is in this century.
- Password validation:
 1. If "New password" or "New password again" has text, show error if they don't match.
 2. If they match, check is it good password: It must have both upper case and lower case letters and numbers. And at least length of 12 characters.
 3. If passwords match and are good passwords, check if "Current password" is correct.
 4. If all of these are correct, change user's password to new one.