



Kanta-asiakassovelluksen toiminnallinen määrittely



Mari, Ek

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Kanta-asiakassovelluksen toiminnallinen määrittely

Ek, Mari
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Toukokuu, 2010

Mari Ek

Kanta-asiakassovelluksen toiminnallinen määrittely

Vuosi 2010 Sivumäärä 81(45+36)

Tämän opinnäytetyön tarkoituksena oli tutkia määrittelyprosessia. Formlek Oy on keksinyt uuden kanta-asiakasformaatin, josta on tarkoitus kehittää tulevaisuudessa sovellus. Sovellusidea on uusi, eikä siitä ole olemassa aikaisempaa versiota. Sovelluksen toimintatarkoituksena on tehostaa ja helpottaa markkinointia.

Tavoitteena oli löytää olennaiset vaatimukset kanta-asiakassovellusformaateista ja muuntaa ne toiminnallisen määrittelydokumentin muotoon. Toiminnallisen määrittelyn tavoitteena oli tuottaa määrittelydokumentti, joka kuvaa toiminnalliset ja ei-toiminnalliset vaatimukset.

Tutkimuksen menetelmänä käytettiin konstruktiivista tutkimusotetta. Teoria tiedon pohjalta luotiin uutta tietoa ja hyödynnettiin sitä kanta-asiakassovelluksen toiminnallisessa määrittelyssä. Työn teoreettinen viitekehys koostuu ohjelmistokehityksen vaiheista, menetelmistä ja työkaluista.

Työn käytännön osuudessa määrittelydokumentti luotiin perustuen IEEE(the Institute of Electrical and Electronics Engineers)-830-standardiin. Tämä standardi määrittelee toiminnalliselle määrittelylle sisällysluettelon. UML (Unified Modeling Language) on graafinen mallinnuskieli, jonka avulla voidaan tehdä kaavioita ja malleja. UML-tekniikkaa hyödynnettiin toiminto- ja käyttötapauskaavioiden luomisessa. Kaavioiden tueksi kirjoitettiin tekstimuotoiset selitykset. Tietokannan määrittelyssä käytettiin relaatiomallia. Käyttöliittymän määrittelyssä ja suunnittelussa käytettiin hyödyksi parhaita käytettävyyden ja käyttäjälähtöisiä menetelmiä.

Lopputuloksena syntyi toiminnallinen määrittelydokumentti ja tavoitteet saavutettiin. Toimeksiantaja hyväksyi dokumentin, minkä jälkeen se versioitiin numeroksi yksi. Dokumenttia on tarkoitus hyödyntää suunnittelussa, toteutuksessa ja testauksessa. Toiminnallinen määrittelydokumentti on liitteessä 1.

Tutkimuksen ymmärsin määrittelyprosessin ja dokumentoinnin merkityksen ohjelmistokehityksessä. Lopputulos onnistui kokonaisuudessaan hyvin ja toimeksiantaja antoi positiivista palautetta tehdystä työstä.

Asiasanat: Määrittely, dokumentointi, UML, tietokanta

Mari Ek

Functional definition of the loyal customer application

Year	2010	Pages	81(45+36)
------	------	-------	-----------

The purpose of this Bachelor's thesis was to research the process of determination. Formlek Oy has invented a new loyal customer model, which is to be developed to an application in the future. The purpose of the application is to streamline and facilitate marketing.

The objective was to create a functional definition of the loyal customer model. The focus was on discovering the relevant requirements in the loyal customer model information. The functional definition document describes the functional and non-functional requirements.

A constructive research method was used in the study. New information was created based on the theory information and it was utilised in the functional definition of the loyal customer model. The theoretical reference of the thesis consists of the stages of the software development, methods and tools.

In the practical section of the thesis the definition document was created based on the IEEE 830 standard. This standard specifies a table of contents to a functional definition. The UML (Unified Modeling Language) language is a graphical modeling language in the field of software engineering. The UML technique was utilised in the creation of a function and use case design. The text format assistance was written to support the design. A relational model was used in the definition of the database. In the definition and planning of the user interface, the best usability and user-centred methods were used.

The functional definition document was completed and the objective was achieved. A functional definition document was created as a final result. The principal accepted the document and it was modified as number one. The purpose is to utilise the document in planning, realisation and testing. The definition document is in Appendix 1.

After the research there was clear understanding of the software development in the definition process and documentation. The final result was successful and the client gave positive feedback on it.

Key words: Determine, documentation, UML, database

Sisällys

1	Johdanto.....	6
1.1	Toimeksiantaja	6
1.2	Tutkimusongelma.....	7
1.3	Aiheen rajaus	8
1.4	Tavoitteet ja tarkoitus	8
1.5	Tutkimuksen ja työn menetelmät.....	9
1.6	Kanta-asiakassovelluksen käsitteitä	11
2	Ohjelmistotuotanto	12
2.1	Ohjelmiston elinkaarimalli	13
2.2	Esitutkimus	14
2.3	Määrittelyprosessi	14
2.4	Toiminnallinen määrittely	15
3	Dokumentointi ja spesifikaatio	16
3.1.1	Dokumentointimallit	17
3.1.2	Toiminnallinen määrittelydokumentti	18
4	UML-mallinnus	19
4.1.1	UML:n käyttäminen.....	20
4.1.2	Menetelmän ja mallinnuksen erot.....	20
4.2	Käyttötapaus	21
4.3	Toimintokaavio	24
5	Tietokanta ja tieto	25
5.1	Relaatiotietokanta	26
5.2	Relaatiotietokannan peruskäsitteet	27
5.3	Perusavain, viiteavain ja toissijainen avain	29
6	Käytettävyys ja käyttöliittymän suunnittelu	30
7	Kanta-asiakassovelluksen määrittelyprosessi.....	32
7.1	Kanta-asiakassovelluksen esitutkimus ja vaatimuskartoitus	32
7.2	Toiminnallisen määrittelydokumentin luominen.....	33
8	Yhteenveto	41
	Kuvat ja kuvat	44
	Taulukot	44
	Liitteet.....	45

1 Johdanto

Opinnäytetyön tarkoituksena on tutkia kanta-asiakassovelluksen määrittelyprosessia ja tuottaa toiminnallinen määrittelydokumentti Formlek Oy:lle. Kanta-asiakasformaatti on uusi, eikä siitä ole olemassa aikaisempaa versiota. Kanta-asiakassovellus perustuu MainosAvain-formaattiin, jonka tekijänoikeudet omistavat Seppo Weckman. Kanta-asiakassovelluksella ei ole vielä virallista nimeä, joten sitä kutsutaan tällä hetkellä Valtuutetuksi markkinoijaksi. Opinnäytetyössä sovelluksesta käytetään määritteenä kanta-asiakassovellus.

Kanta-asiakassovelluksen määrittelyssä sovelletaan ohjelmistokehittämisen menetelmiä ja työkaluja. Työn lisäarvoa kasvatetaan tutkimalla ohjelmistokehityksen kirjallisuutta ja lähteitä. Toiminnallisen määrittelyn tuloksena syntyy toimeksiantajalle määrittelydokumentti. Tutkimuksessa kerrotaan tuloksien tekemisen ja syntyminen vaiheista. Tutkimuksen viimeisessä luvussa analysoidaan tutkimuksen onnistumista kokonaisuutena.

1.1 Toimeksiantaja

Formlek Oy on virtuaalisia mainoksia kehittävä yritys. Yrityksen toiminta-ajatuksena on tuottaa innovatiivisia, joustavia, yksinkertaisia yhteistyö-, mainos- ja markkinointiratkaisuja, mitkä perustuvat MainosAvain-formaattiin. Yksinkertaiset ratkaisut takaavat varman toiminnan ja edulliset palveluhinnat.

Palvelut on suunniteltu yrityksille, joiden arvoihin kuuluu verkostoituminen. Osa palveluista soveltuu myös kuluttajille.

Ratkaisut perustuvat MainosAvain-formaattiin, joka tunnetaan tavaramerkkien ja hakukoneiden sekä tietorakenteiden kokonaishallinnasta. Ratkaisut tuotetaan yhdessä alan taitajien kanssa.

Formlek Oy:n arvot ovat:

- Tehdään toimivaa
- Tehdään yhdessä

- Tehdään paremmin.

Yrityksen yhteyshenkilönä toimii Seppo Weckman. Toimitilat sijaitsevat Helsingissä Lauttasaaren Pienteollisuustalossa. Yrityksen kotisivut löytyvät osoitteesta: www.mainosavain.fi. Formlek Oy ja Laurea-ammattikorkeakoulu tekevät yhteistyötä, minkä johdosta opiskelijoilla on mahdollisuuksia tehdä opinnäytetöitä tai hankkeita. (MainosAvain 2010.)

1.2 Tutkimusongelma

Opinnäytetyön aiheena on tutkia kanta-asiakassovelluksen määrittelyprosessia. Formlek Oy on kehittänyt innovatiivisen formaatin kanta-asiakassovelluksen, joka perustuu MainosAvain-formaattiin. MainosAvain-formaatista löytyy lisätietoa Internet-sivustolta: <http://www.mainosavain.fi/>.

Kanta-asiakassovelluksen perusajatuksena on yhdistää perinteisen kaupan sekä verkkokaupan hyvät puolet ja vähentää niiden heikkoja puolia. Kanta-asiakassovelluksen avulla on tarkoitus tehostaa markkinointia, löytää oikeat markkinoinnin kohderyhmät, ylläpitää asiakasrekisteriä ja automatisoida tarjouksien hallintaa. Uuden sovelluksen toimintoja ovat kanta-asiakkaaksi rekisteröityminen, asiakastietojen päivittäminen, tarjouspyyntöjen ja tarjouksien lähettäminen. Kanta-asiakassovellusta pitäisi pystyä käyttämään Web-selaimella.

Tutkimusongelman pääkysymyksenä on: miten kanta-asiakasformaattista luodaan toiminnallinen määrittelydokumentti? Sen jälkeen tutkimuskysymys ositettiin alakysymyksiin:

- Mitä on määrittelyprosessi tarkoittaa ja mitä asioita siihen kuuluu?
- Miten ohjelmistokehitysprosessi etenee kokonaisuutena?
- Mitä ohjelmistokehitysprosessin menetelmiä ja työkaluja voidaan käyttää määrittelyprosessissa?
- Mikä on dokumentoinnin merkitys ohjelmistokehityksessä ja miten toiminnallinen määrittelydokumentti luodaan?
- Mikä on esitutkimus ja miten se liittyy toiminnallisen määrittelydokumentin tuottamiseen?

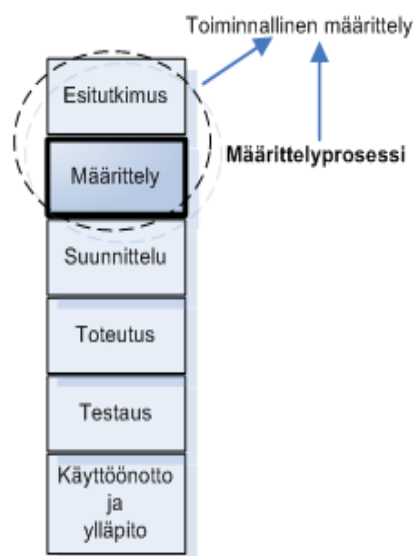
Opinnäytetyössä etsitään vastauksia tutkimusongelmiin, joihin on tarkoitus löytää hyödyllisiä ja käyttökelpoisia ratkaisuja ICT-alan kirjallisuudesta. Kirjallisuutta ja Internetlähteitä pyritään hyödyntämään määrittelyprosessin tutkimisessa ja toiminnallisen määrittelydokumentin luomisessa. Työn tarkoituksena on ratkaista toimeksiantajan ongelma ja käyttää erilaisia menetelmiä apuna.

1.3 Aiheen rajaus

Opinnäytetyön ulkopuolelle rajataan laitteistojen, komponenttien, tietoliikenneyhteyksien ja Web-hotellin suunnittelu ja hankinta.

Ohjelmistoprosessista rajataan pois kanta-asiakassovelluksen tekninen suunnittelu, sovelluksen toteuttaminen C# -ohjelmointikielellä, testaus, käyttöönotto ja ylläpito (kuva 1).

Tutkimuksen rajaus:



Kuva 1: Tutkimustyön rajaus

1.4 Tavoitteet ja tarkoitus

Opinnäytetyön tavoitteena on tutkia kanta-asiakassovelluksen toiminnallisen määrittelyn kehitysprosessia. Tutkimuksen tavoitteena on etsiä tietoa ohjelmistokehityksen vaiheista, menetelmistä ja toiminnallisen määrittelyn

merkityksestä osana kanta-asiakassovelluksen kehitystä. Tavoitteena on löytää olennaiset vaatimukset kanta-asiakassovellusformaatista ja muuntaa ne toiminnallisen määrittelydokumentin muotoon. Tutkimuksen tarkoituksena on tuottaa lisäarvoa kanta-asiakassovelluksen toiminnallisen määrittelydokumentin luomiseen.

Toiminnallisen määrittelyn tavoitteena on tuottaa määrittelydokumentti, joka kuvaa määrittelyprosessiin kuuluvat toiminnalliset ja ei-toiminnalliset vaatimukset. Toiminnallisen määrittelydokumentin keskeisin tavoite on esittää käsitystä, siitä millainen on toteutettava kanta-asiakassovellus. Toiminnallista määrittelydokumenttia on tarkoitus hyödyntää kanta-asiakassovelluksen suunnittelussa ja toteutuksessa. Työn tavoite on saavutettu, kun toimeksiantaja hyväksyy toiminnallisen määrittelydokumentin. Hyväksynnän jälkeen dokumentti versioidaan numeroksi yksi (1.0). Määrittelydokumentin tarkoituksena on vähentää suunnittelussa ja toteutusvaiheessa tulevia muutoksia sekä ennustaa tulevaa työmäärää.

Työn tekijän henkilökohtaisena tavoitteena on kasvattaa oman alan asiantuntijuutta, oppia uutta, ymmärtää määrittelyprosessin merkitys, tutkia ohjelmistotuotannon kehittämismenetelmiä ja kehittää ongelman ratkaisutaitoja.

1.5 Tutkimuksen ja työn menetelmät

Tutkimusmenetelmäksi on valittu konstrukttiivinen tutkimusote. Tutkimuksen keskeiset elementit esitellään kuvassa 2. Tutkimuksen avulla etsitään ongelmaan ratkaisua eli konstruktiota. Tutkimuksen aihe kuuluu suunnittelutieteen alueelle, koska se sisältää uuden ohjelmiston arviointia, rakentamista ja määrittelyä. Tutkimuksessa pyritään ratkaisemaan toimeksiantajan ongelma etsimällä aiheeseen liittyvää tutkimustietoa. Aikaisempaa tutkimustietoa soveltaen testataan ratkaisun toimivuutta käytännössä ja luodaan sen pohjalta uutta tietoa. Lopuksi arvioidaan mikä on tutkimusongelman ja kehitetyn ratkaisun käytännöllinen merkitys. (Lukka 2001.)

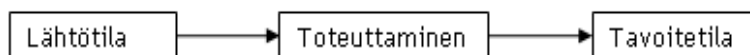


Kuva 2: Konstruktiivisen tutkimusotteen keskeiset elementit

Konstruktiivinen tutkimusote sisältää käytäntöön suuntautuneen totuuskäsityksen, jossa perustellaan, että se mikä toimii, on totta. Käytännössä pragmaattinen totuustesti tulee todennettua, kun kehitettyä toiminnallista määrittelydokumenttia hyödynnetään kanta-asiakassovelluksen suunnittelussa ja toteutuksessa. Ratkaisussa pyritään yksinkertaisuuteen, helppokäyttöisyyteen ja sen merkityksellisyyteen. Kuten aiemmin jo mainittiin tullaan tutkimustyön lopussa kertomaan konstruktion tuloksista. (Seppänen 2004.)

Järvisen & Järvisen (2004, 103-104) mukaan, jos tutkimusongelma sisältää konstruktiiviseen tutkimusmenetelmään liittyviä verbejä esimerkiksi rakentaa, muuttaa, parantaa, vahvistaa, huoltaa, laajentaa, korjata, sovittaa, laatia, kuuluu työ suunnittelutieteen alueelle.

Toteutusprosessi etenee kuvan 3 mukaan. Työn lähtötilanteena on tutkimusongelma. Lähtötilanteessa tutustutaan aiheeseen ja etsitään teoretietoa. Sen jälkeen pohditaan, millaisia menetelmiä ja työkaluja tarvitaan tavoitteen saavuttamiseksi. Tavoitetila on saavutettu siinä vaiheessa, kun toimeksiantajalle on luotu kanta-asiakassovelluksen määrittelydokumentaatio. Dokumentaation täytyy olla hyväksytty ja versioitu numeroksi yksi.



Kuva 3: Toteutusprosessi

Tärkeimmät menetelmät toteutusvaiheessa ovat määrittelyprosessi, dokumentointi- ja relaatiomallit. Kuvauksien luomisessa käytetään apuvälineenä UML-mallinnusta. Määrittelydokumentin kuvaukset toteutetaan Microsoft Visio 2003 Pro-ohjelmistolla. Dokumentointi toteutetaan IEEE-830-standardia soveltaen ja hyödyntäen MS Officen Word-ohjelmaa. Sovelluksen käyttöliittymän toiminnoista luodaan muutama protomalli ja kuvataan alustavat komponentit. Käyttöliittymän suunnittelussa otetaan huomioon käyttäjälähtöisyyden ja käytettävyyden hyväksi havaittuja menetelmiä.

1.6 Kanta-asiakassovelluksen käsitteitä

Kanta-asiakas:

Kanta-asiakas on kanta-asiakassovelluksen käyttäjä, joka etsii kiinnostavia ja edullisia tuotteita sovelluksen avulla. Asiakas voi tehdä tarjouspyyntöjä ja vastaanottaa tarjouksia myyjiltä. Asiakas hallitsee kanta-asiakassivuston kautta omia tietojaan, mm. henkilö- ja profiilitietoja.

Myyjä:

Myyjä markkinoi ja myy tuotteita. Myyjät voivat ottaa yhteyttä asiakkaisiin sallituilla markkinointikeinoilla. Asiakasprofiilin avulla myyjät voivat kohdistaa oikeita tuotteita asiakkaalle.

Sallittu markkinointitapa:

Sallitulla markkinointitavalla tarkoitetaan sitä keinoa, miten myyjä voi markkinoida tuotetta asiakkaalle. Markkinointimuodot ovat valtion posti, sähköposti, soitto, sähköinen mainos tai tekstiviesti.

Tarjous:

Tarjous sisältää tarjouksen nimen, hinnan ja voimassaolo päivän. Tarjouksen voimassaolo ajat ovat mm. päivä, viikko tai kuukausi.

Profiili:

Asiakas pystyy omassa profiilissa muokkaamaan profiilitietoja. Profiilitiedoissa hallitaan sallittuja markkinointitapoja ja kiinnostuksen kohteita. Profiilia voidaan muokata milloin vain ja tiedot päivittyvät välittömästi tallentamisen jälkeen.

Tarjouspyyntö:

Asiakas jättää myyjälle tarjouspyynnön, joko liikkeessä virtuaalisella kanta-asiakaskortilla tai sovelluksen kautta liittymällä eri kategorioihin. Toistaiseksi myös sovelluksen yli voidaan jättää tarjouspyyntöjä, esim. sähköpostilla tai suullisesti.

Kategoria (suosikit):

Asiakkaat voivat liittyä erilaisiin kategorioihin. Kategoriat ovat kiinnostuksen kohteita mm. atk, elokuvat, matkat, pelit, taide, harrastukset, urheilu, sisustaminen, tähtitiede jne.

Sähköinen mainos / e-mainos / e-tarjous:

HMTL -pohjainen sähköinen mainos, mikä voidaan lähettää sivustolle kanta-asiakassovelluksen kautta.

Virtuaalinen kanta-asiakaskortti + tunnus:

Virtuaalinen kanta-asiakaskortti koostuu puhelinnumerosta + tunnuksesta. Kanta-asiakaskortin voi tulostaa myös viivakoodina. Liikkeessä asiakas voi jättää yhteydenottopyynnön antamalla kanta-asiakastiedot mm. kanta-asiakaskortilla (viivakoodina) ja syöttämällä näppäimistöllä tunnuksen. Tiedot voidaan antaa myös suullisesti.

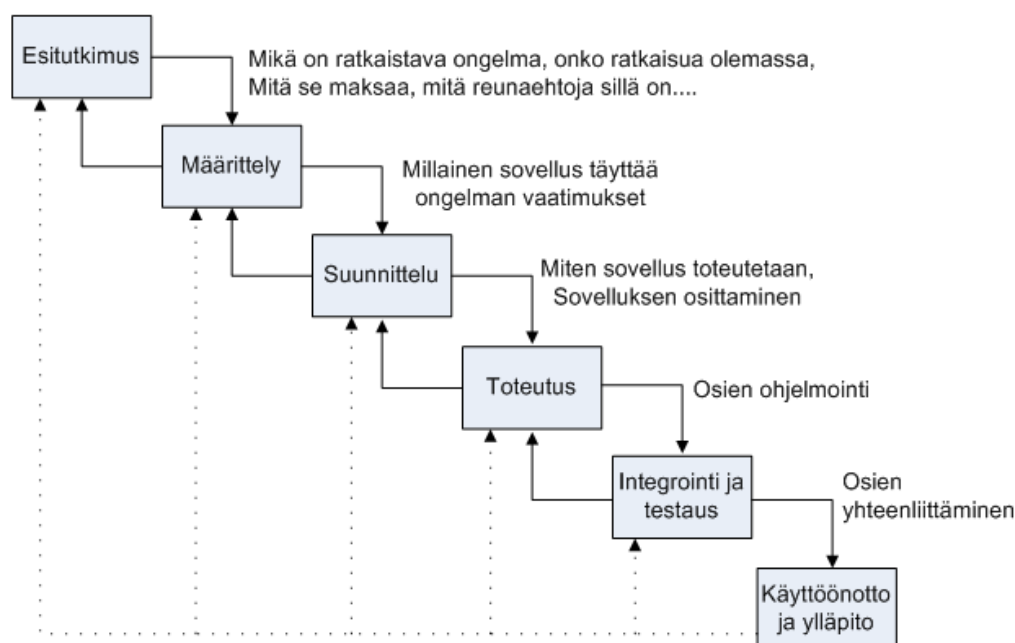
2 Ohjelmistotuotanto

Ohjelmistotuotanto voidaan jakaa osa-alueisiin. Kehitysprosessissa voidaan eritellä seuraavat vaiheet: määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. Tukitoimintoja ohjelmistokehityksessä ovat laadunvarmistus, tuotteenhallinta ja dokumentointi. Jokaiseen osa-alueeseen liittyvät omat käsitteistönsä, erityisongelmasa, menetelmänsä ja työkalunsa. (Haikala & Märijärvi 2002, 35-36.)

2.1 Ohjelmiston elinkaarimalli

Ohjelmiston elinkaarella (life cycle) tarkoitetaan kuluvaan aikaan, joka kuluu ohjelmiston kehittämisen alusta lopulta sen poistamiseen käytöstä. Vaihejaolla tarkoitetaan tapaa, jolla ohjelmistotyö voidaan jakaa eri vaiheisiin. Kuvassa 4 on esimerkki yhdestä vesiputousmallin versiosta (waterfall model), joka on tavallisin ja yleisin käytetty vaihejakomalli. Mallista on olemassa eri variaatioita, mutta yleensä niissä esiintyvät ainakin määrittely, suunnittelu ja toteutusvaiheet. Muita käytettyjä vaihejakomalleja ovat mm. Evo-malli (evolutionary delivery) ja eri protomallit. Kanta-asiakassovelluksen kehittäminen noudattaa vesiputousmallin mukaista elinkaarimallia. Elinkaarimallin vaiheista tutkimus keskittyy määrittelyprosessiin.

Ennen määrittelyvaihetta on yleensä syytä tutkia rakennettavan ohjelmiston asiakasvaatimuksia eli tätä vaihetta kutsutaan esitutkimukseksi. Esitutkimuksen jälkeen määrittelyvaiheessa eritellään asiakasvaatimuksia ja niistä tehdään ohjelmistovaatimukset, jotka määrittelevät kehitettävän ohjelmiston. Määrittelyn tuloksena syntyy toiminnallinen määrittelydokumentti. Toiminnallisesta määrittelystä kerrotaan lisää seuraavassa luvussa. (Haikala & Märijärvi 2002, 37-38.)



Kuva 4: Perinteinen vesiputousmalli

Suunnitteluvaiheessa toteutetaan määrittelyn kuvaksen pohjalta suunnitteluvaihe. Suunnitteluvaihe voidaan jakaa kahteen vaiheeseen: arkkitehtuurisuunnittelu ja moduulisuunnittelu. Arkkitehtuurin tarkoituksena on jakaa ohjelmisto mahdollisimman itsenäisiin, toisista riippumattomiin osiin eli moduuleihin. Arkkitehtuurin tuloksena syntyy tekninen määrittelydokumentti. Moduulisuunnittelun tarkoituksena on suunnitella jokaisen moduulin sisäinen rakenne. Moduulilla tarkoitetaan loogista kokonaisuutta, joka sisältää yleensä alle 1000 koodiriviä. (Haikala & Märijärvi 2002, 39-40.)

Toiminnallisen määrittelyn jälkeen on tarkoituksena siirtyä kanta-asiakassovelluksen suunnitteluvaiheeseen. Suunnitteluvaiheen jälkeen toteutusvaiheessa toteutetaan kanta-asiakassovellus C#-ohjelmointikielellä ja luodaan tietokanta.

Toteuttamisen jälkeen etsitään ohjelmistosta virheitä testauksella ja tehdään tarvittavat korjaukset. Testauksen yleisin laadunvalvonnan toiminto on suoritustestaus eli vikojen löytäminen suorittamalla sovellus ja tarkkailemalla, mitä se tekee. (McConnell 2002, 72).

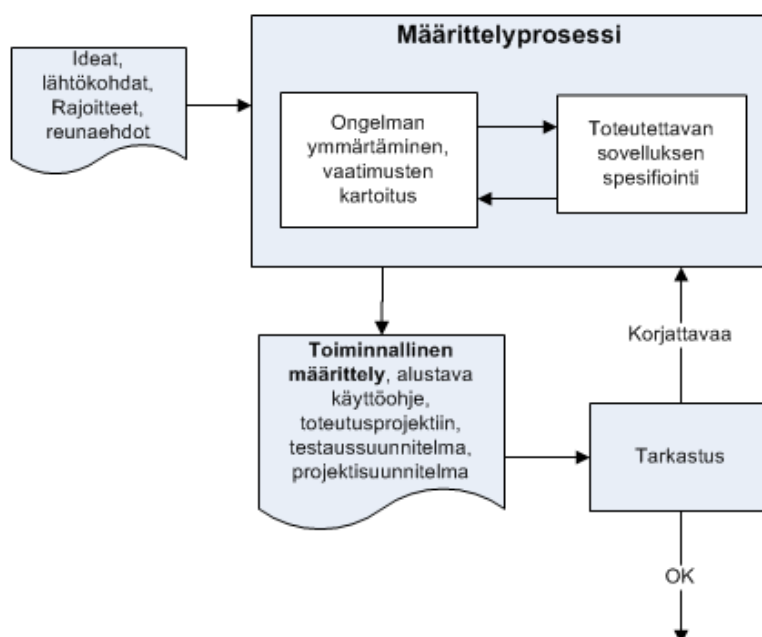
2.2 Esitutkimus

Esitutkimuksen tarkoituksena on selvittää, onko ohjelmiston rakentaminen mahdollista. Esitutkimuksessa ei vielä rakenneta mitään, eikä toteuteta mitään ja tehdä teknisiä ratkaisuja. Ajatuksena on selvittää ensin ohjelmiston tekemisen syyt, tavoitteet ja millaisia ratkaisuvaihtoehtoja on olemassa. (Pohjonen 2002, 27.) Esitutkimustietoja käydään läpi toimeksiantajan kanssa yhteisissä tapaamisissa ja sähköpostin välityksellä. Vaatimukset ja tarpeet huomioidaan toiminnallisessa määrittelyssä. Näiden tietojen pohjalta on tarkoituksena luoda toteutettavan sovelluksen toiminnallinen määrittely.

2.3 Määrittelyprosessi

Määrittelyyn liittyvät tehtävät ovat projektin tarpeellisuuden ja toteuttamiskelpoisuuden selvittäminen, tavoitteiden ja vaatimusten asettaminen sekä ratkaisumallien laatiminen. Määrittelyprosessi on kuvattu kokonaisuutena kuvassa 5. Tärkeää on ymmärtää ongelman ydin ja löytää oikeat vaatimukset. Määrittelyvaihe jaetaan yleensä kahteen osaan: asiakasvaatimuksien analysointiin ja

kehittävän sovelluksen määrittelyyn. Asiakasvaatimuksien kartoittamista kutsutaan esitutkimukseksi. Määrittelyprosessin lopputuloksena syntyy toiminnallinen määrittelydokumentti. Tässä tutkimuksessa ei tehdä varsinaisesti esitutkimus raporttia, mutta pyritään huomioimaan kaikki tarpeet ja vaatimukset. (Haikala & Märijärvi 2002, 78-79.)



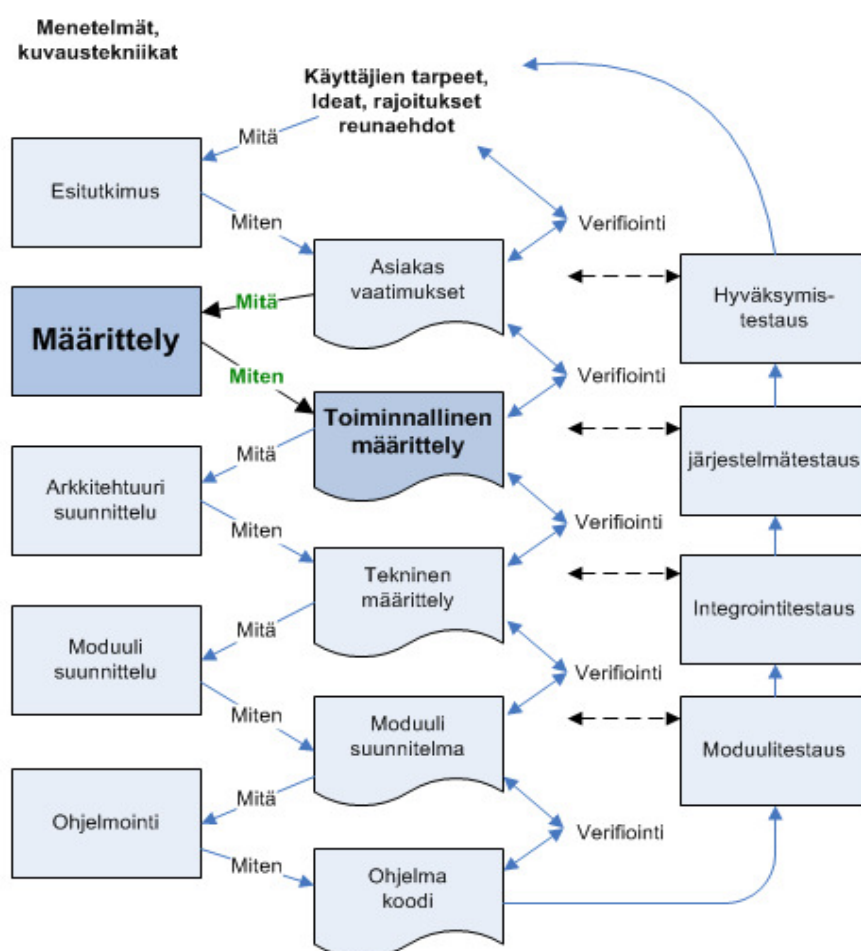
Kuva 5: Määrittelyprosessi

2.4 Toiminnallinen määrittely

Toiminnallisen määrittelyn tarkoituksena on selvittää, millaista sovellusta ollaan kehittämässä ja miten se tehdään. Määrittelyprosessin tärkeimpänä lopputuloksena syntyy toiminnallinen määrittelydokumentti. Toiminnallisen määrittelyn tarkoituksena on kuvata ohjelmiston toiminnot, toteutukselle asetettavat ei-toiminnalliset vaatimukset sekä rajoitukset. Toiminnallisia vaatimuksia ovat esimerkiksi, millaisessa ympäristössä ohjelmisto tulee toimimaan ja sen vaatimat toiminnot. Ei-toiminnalliset vaatimukset ovat esimerkiksi suoritusteho, vasteaika ja käytettävyys. Rajoituksia ovat muistintila, toteutus tietyllä ohjelmointikielellä jne. Tässä määrittelyvaiheessa pyrkimyksenä on muuttaa asiakasvaatimukset ohjelmistovaatimuksiksi. (Haikala & Märijärvi 2002, 38-39.)

3 Dokumentointi ja spesifikaatio

Ohjelmistotuotanto on spesifikaatioiden laatimista, mikä on teknisestä näkökulmasta dokumenttien tuottamista. Spesifikaatio on vaatimuksia asetettava kuvaus jostakin teknisestä aiheesta. Spesifikaatiota kutsutaan speksiksi. Tuotantoprosessi on (ks. kuvassa 6) vaiheittain kuvattua tuottamista sovelluksen vaatimuksista niiden toteuttamiseksi. Määrittely ja toiminnallinen määrittelylaatikot ovat kuvassa 6 korostettuna, koska ne keskittyvät tämän tutkimuksen aihealueeseen. Juuri tällaisten speksien tuottamiseen tarvitaan erilaisia kuvaustekniikoita ja dokumentteja. Määrittelyvaiheen syötteenä ovat asiakasvaatimukset ja sen seurauksena syntyy toiminnallinen määrittely -dokumentti. (Haikala & Märijärvi 2002, 61.)



Kuva 6: Tuotantoprosessi

Yleensä määrittely vastaa kysymykseen ”mitä” ja suunnittelu kysymykseen ”miten”. Määrittelyllä ei rasiteta lukijaa ohjelmiston toteutuksen yksityiskohdilla vaan kuvataan se käyttäjän kannalta ratkaistuna ongelmana. Hyvän spesifikaation ominaisuuksia ovat täydellisyys, tarkkuus, virheettömyys, ymmärrettävyys, testattavuus ja jäljitettävyys. Tässä tutkimuksessa käytännön osassa pyritään tuottamaan hyvä spesifikaatio. (Haikala & Märijärvi 2002, 62-65.)

Ohjelmistoprojekteissa on yleistä, että siinä kirjataan työvaiheiden aikana tullutta tietoa dokumenttien muotoon. Isoissa projekteissa syntyy tietysti monia kymmeniä dokumentteja, kun taas pienissä ohjelmistoprojekteissa selvittää vähemmällä dokumenttimäärällä. Suositeltuja dokumentaatioita ovat: projektisuunnitelma, määrittelydokumentti, suunnitteludokumentti ja testaussuunnitelma. Ylläpidon kannalta on syytä aina päivittää dokumentteja, jos niihin tulee muutoksia. (Haikala & Märijärvi 2002, 50-51.)

3.1.1 Dokumentointimallit

Dokumentteja varten kannattaa tehdä dokumentointimallit. Dokumenttiin tehtäessä CASE-välineillä piirrettyjä kaavioita, tulee niihin tehdä asianmukaiset selitykset. Hyvä dokumentointimalli sisältää seuraavat perusasiat:

- kansilehti
- sisällysluettelo
- päivämäärä
- organisaation nimi
- projektin nimi
- dokumentin nimi
- versionumero
- sivujen määrä
- tekijät
- tila (hyväksytty, jätetty tarkastettavaksi, keskeneräinen).

3.1.2 Toiminnallinen määrittelydokumentti

Hyvä esimerkki toiminnallisen määrittelydokumentin sisältörungosta on kuvassa 7. Sisältö on vapaasti muotoiltu IEEE(Institute of Electrical and Electronics Engineers) 830-1998-standardista. Dokumentin johdantokappaleessa selvitetään, miksi tuote on tehty ja kenelle se on tarkoitettu ja mikä on sovelluksen tavoite (kuvan kohdat 1.1 ja 1.2). Kohdassa 1.3 voidaan määritellä tarvittaessa sovelluksen määritteitä, teknisiä määritteitä sekä lyhenteitä. Kohta 1.4 kannattaa kuvata, jos asiaan liittyviä dokumentteja on olemassa. Ensimmäisen osan viimeisenä kappaleena kannattaa esittää yleiskatsaus dokumentin rakenteeseen. Katsauksen tarkoituksena on antaa tietoa siitä, mitkä kohdat hänen kannattaa lukea. (Haikala & Märijärvi 2002, 79-80.)

Kappaleessa kaksi esitetään yleiskuva järjestelmän toiminnasta. Yleiskuvauksen kohdissa 2.1 kuvataan järjestelmän liittymät sen ympäristöön ja kuvataan myös ympäristöä sopivalla tarkkuudella. Kohdassa 2.2 Kerrotaan yleiskuva järjestelmän toiminnoista. Kohdassa 2.3 kuvataan järjestelmän käyttäjiä. Kohdassa 2.4 kuvataan yleiset rajoitteet esim. standardit ja lainsäädäntö. Luvun viimeisessä kohdassa 2.5 esitetään oletukset ja riippuvuudet.

Kappaleessa kolme kuvataan järjestelmän tiedot ja tietokanta, sekä niiden tietosisältö. Kappaleessa neljä täsmennetään jokainen järjestelmän toiminto. Kaikista toiminnoista esitetään sen tarkoitus, mitä toiminto tarvitsee syötteinään, miten käsittely tapahtuu ja mitä vaikutuksia toiminnolla on eli mitä tulosteita saadaan aikaiseksi.

Kappaleessa viisi tarkoituksena on täsmentää aiemman luvun 2.1 yleisesti kuvatut liittymät järjestelmän toimintaympäristössä. Käyttöliittymän kuvaus voidaan tehdä jo tässä vaiheessa tarkasti tai esittää se yleisellä tasolla. Muita esitettäviä asioita ovat tietoliikenne yhteydet, oheislaitteet ja ohjelmistoliittymät. Luvut kuusi ja seitsemän sisältävät ei-toiminnalliset vaatimukset, kuten suorituskyky, turvallisuus, ylläpito, rajoitteet ja standardit. (Haikala & Märijärvi 2002, 80-81.)

1.	Johdanto	5.	Ulkoiset liittymät
1.1	Tarkoitus	5.1	Käyttöliittymä
1.2	Tuote	5.2	Laitteistoliittymät
1.3	Määritelmät, termit ja lyhenteet	5.3	Ohjelmistoliittymät
1.4	Viitteet	5.4	Tietoliikenneliittymät
1.5	Yleiskatsaus dokumenttiin	6.	Muut ominaisuudet
2.	Yleiskuvaus	6.1	Suorituskyky
2.1	Ympäristö	6.2	Käytettävyys, toipuminen, turvallisuus ja suojaukset
2.2	Toiminta	6.3	Ylläpidettävyys
2.3	Käyttäjät	6.4	Siirrettävyys, yhteensopivuus
2.4	Yleiset rajoitteet	6.5	Operointi
2.5	Oletukset ja riippuvuudet	7.	Suunnittelu rajoitteet
3.	Tiedot ja tietokanta	7.1	Standardit
4.	Toiminnot	7.2	Laitteistorajoitteet
4.1	Toiminnon kuvaus	7.3	Ohjelmistorajoitteet
		7.4	Muut rajoitteet

Kuva 7: Toiminnallisen määrittelyn sisältörunko-IEEE 830

Dokumenttia kuvitetaan eri notaatioilla tehdyillä kuvauksilla. Näistä kuvaus tavoista kerrotaan lisää seuraavassa luvussa 4 UML-mallinnus.

4 UML-mallinnus

UML (modeling language) on mallinnuskieli, mitä voidaan hyödyntää määrittelyssä ja suunnittelussa. Mallinnuskieli on pääosin graafinen kuvauskieli, jota kutsutaan notaatioksi. Notaation avulla voidaan kuvata tuloksen muotoa ja toimintoja. Mallinnustapa on tärkeä väline ihmisten välisessä viestinnässä. Sen avulla voidaan esittää visuaalisesti järjestelmän rakennetta toiselle henkilölle. UML-kielen avulla voidaan esittää ja selventää käsitteitä selkeämmin. (Eriksson-Penger 2000, 4-5).

Mallintaessa täytyy olla selvillä tuotteelle asetuista vaatimuksista. Vaatimukset voivat olla seuraavanlaisia ominaisuuksia kuten, toiminnallisuus, ulkonäkö, tehokkuus tai luotettavuus. Mallin tekeminen on luovuutta vaativaa työtä. Mallinnuksessa parhaat lopputulokset saavutetaan yleensä runsaalla ideoinnilla, jossa kuvataan ja kokeillaan useita näkemyksiä. Eri vaihtoehtoja kehittämällä suunnittelijat oppivat ymmärtämään järjestelmään paremmin ja oivat lopulta kehittää mallit, jotka kuvaavat järjestelmän ja sen käyttäjien asettamat tavoitteet ja vaatimukset.

Käyttökelpoinen malli on tarkka, yhtenäinen, helposti muokattavissa ja ymmärrettävää. (Eriksson-Penger 2000, 1-5.)

4.1.1 UML:n käyttäminen

UML-mallinnusta voidaan käyttää laajasti monien erityyppisten järjestelmien kuvaamiseen. UML:ää voidaan käyttää tietojärjestelmien kehittämisen eri vaiheissa. Eniten UML- mallinnusta käytetään ohjelmistokehityksen parissa, mutta sitä voidaan käyttää myös mekaanisten laitteiden tai yrityksen toiminnan kuvaamiseen.

4.1.2 Menetelmän ja mallinnuksen erot

Menetelmän ja mallinnuskielen välillä on tärkeitä eroja. Menetelmä on tietty tapa järjestellä ajatuksia ja toimintoja. Menetelmä kertoo, mitä pitää tehdä, kuinka se tehdään, milloin se tehdään ja miksi se tehdään, se kertoo siis toiminnan tarkoituksen. Menetelmään kuuluu malleja, jotka kuvaavat jotakin ja näyttävät työmenetelmän seuraamisen tulokset. Menetelmän ja mallinnuskielen tärkein ero on se, että mallinnuskielessä ei ole toimintaohjeita. Selvemmin kerrottuna ei ole siis ohjeita sille, mitä, kuinka, milloin ja miksi jotain pitää tehdä.

Mallilla on aina joku tarkoitus jostakin asiasta ja se voidaan kuvailla mallinnuskielen avulla. Mallinnuskieli koostuu notaatiosta eli mallissa käytettävistä merkinnöistä sekä säännöistä. Mallilla pitää olla selvä käyttötarkoitus, jotta tiedetään kuinka ja miksi mallia käytetään. Kielioppisäännöt kertovat, miltä symbolit näyttävät ja kuinka niitä voi mallissa yhdistellä. Käyttösäännöt määrittävät symbolien aikeet, joiden kautta mallin tarkoitus toteutuu ja on muidenkin ymmärrettävissä. Nämä säännöt on opeteltava hyvin, jotta mallinnuskielen käyttö on mahdollisimman tehokasta. (Eriksson-Penger 2000, 6.)

Kaaviot ovat kuvioita, joissa mallinnuselementit on järjestetty kuvaamaan tiettyä järjestelmän piirrettä. Järjestelmän malli koostuu yleensä useista erilaisista kaavioista. Kaavio kuuluu tiettyyn näkymään. Kaavion sisällöstä riippuen jotkut kaavioista voivat kuulua useaan näkymään. Kuvausmenetelmiä ovat esimerkiksi toimintojen ja tietojen kuvaus, luokkakaaviot, tilakaaviot, tapahtumasekvenssien kuvaukset ja tietovirtakaaviot. UML on siis laaja ja monipuolinen kokoelma erilaisia kuvaustekniikoita. (Eriksson-Penger 2000, 14.)

4.2 Käyttötapaus

Käyttötapauksia käytetään ratkaisuna käyttäjävaatimusten kartoittamiseen. Se on yleisin tapa kuvata järjestelmälle asetetut vaatimukset. Käyttötapausten idea on Haikalan & Märijärven mukaan yksinkertainen. Tarkoituksena on kuvata järjestelmän toiminnallisuus joukkona järjestelmän käyttäjien, sillä suorittamia tapahtumaketjuja. Käyttötapaus voi olla esimerkiksi lainaakirja. Käyttötapauksia voidaan kuvata kahdella tavalla joko tekstimuotoisena tai sitten use case-kaaviolla. Käyttötapaus kuvaa seuraavia asioita:

- Toimintaa tai toiminnan osaa
- Tapahtumaa johon pitää reagoida
- Käyttäjän ja tietojärjestelmän välistä kommunikointia
- Käyttötapaus voi olla pieni yksityiskohta tai laaja kokonaisuus

Jokaiseen käyttötapauskaavioon kuuluu yksi tai useampia käyttäjärooleja eli toimijoita (actor) joka suorittaa käyttötapausten. Järjestelmän käyttötapaukset ja käyttäjäroolit kuvataan käyttötapauskaaviossa (use-case-diagram). Toimija voi olla esimerkiksi ylläpitäjä, assistentti, vastuhenkilö, myyjä tai toinen järjestelmä, kuten laskenta järjestelmä. Samalla käyttäjällä voi olla käytännössä useampi kuin yksi rooli. Toimijat kuvataan yleensä tikku-ukkoina. Käyttäjät syöttävät tietoa järjestelmään, eivät ole siis järjestelmän osia. Edellä mainittuun käyttötapaukseen lainaakirja on siis kuuluttava joku toimija ja se voi olla esimerkiksi Lainaaaja. Käyttötapauksessa käynnistäjä voi olla lainaaja, mistä seuraa toiminta eli use case lainaakirja. (Haikala-Märijärvi 2002, 156.)

Käyttötapaus alkaa aina jonkin roolin aloitteesta ja päättyy siihen, että käyttäjä on saanut jonkin mielekkään tehtäväkokonaisuuden suoritetuksi. Hyvän käyttötapausten ominaisuuksia ovat:

- Ymmärrettävyys
- Asiakasvaatimuksia kuvaava
- Testattavuus
- Koko

- Sopiva tarkkuus

Ymmärrettävyydellä tarkoitetaan sitä, että käyttötapaukset kirjoitetaan asiakkaalle ja tulevan järjestelmän käyttäjille ymmärrettävään muotoon. Kuvauksen pitää olla mahdollisimman konkreettinen. Ilmaistaan käyttötapaukset mahdollisimman yksiselitteisesti, ei siis liian tarkasti. Käyttötapauksen koko ei saa olla liian laaja. Haikalan & Märijärven mukaan sopivan kokoinen käyttötapaus on yksi A4-arkki. Käyttötapaukset kattavat tärkeimmät osat toteutuksessa. Kaikkia yksityiskohtia ei pidä ottaa mukaan käyttötapauksiin. (Haikala-Märijärvi 2002, 157).

Miten löytää parhaiten käyttötapauksia? Ensin pitää löytää toimijat ja käyttötapauksen on aina oltava kytköksissä yhteen toimijaan. Käyttötapauksia voidaan etsiä seuraavien kysymysten avulla:

- Mitä toimintoja toimija tarvitsee järjestelmältä? Mitä toimijan pitää saada tehdyksi?
- Täytyykö toimijan lukea, luoda, poistaa, muokata tai tallentaa tietoa järjestelmässä?
- Täytyykö toimijalle kertoa jostakin järjestelmän tapahtumista vai täytyykö toimijan ilmoittaa järjestelmälle jostakin tapahtumasta? Mitä tapahtumat tarkoittavat toiminnallisuuden kannalta?
- Voisiko toimijan päivittäistä työtä helpottaa tai tehostaa uusilla toiminnoilla? Voisiko manuaalista työtä jotenkin automatisoida?

Muita kysymyksiä, joita ei kohdisteta yhdelle toimijalle:

- Mitä syötettä järjestelmä tarvitsee ja mitä tulosteita sen täytyy tehdä? Mistä syötteet tulevat ja minne tulosteet menevät?
- Mitkä ovat nykyisen toteutuksen suurimmat ongelmat, kuten manuaalinen työ automaation sijaan?

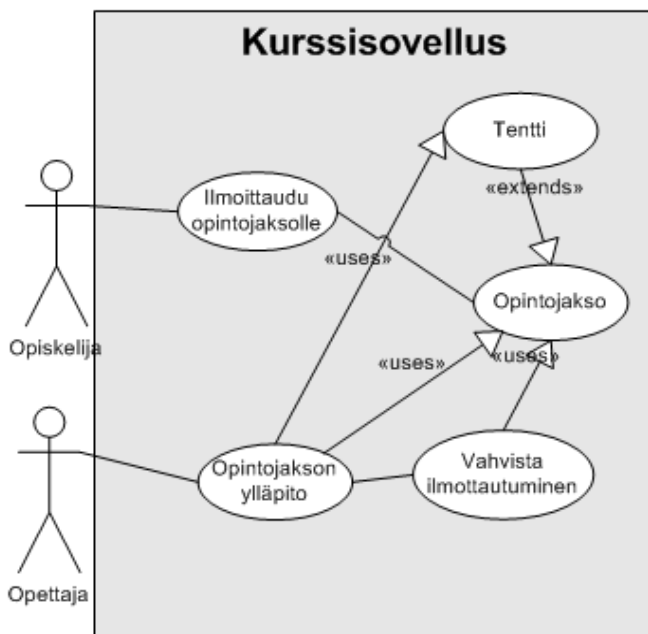
(Eriksson & Penger 2000, 46.)

Tekstikuvauksen sisältö on kuvaus, joka sisältää tiedon siitä, mistä toiminnosta on kyse, mitkä toimijat siihen kuuluvat, kuinka usein se suoritetaan, mitä esiehtoja toimintoon liittyy, mikä on toiminnon yksityiskohtaisempi kuvaus, millaisia

lopputuloksia toiminnon suorituksesta seuraa sekä mitä poikkeuksia saattaa toiminnossa esiintyä.

Käyttötapaustunnus: UC-1 (=Use Case-1)	
Käyttötapauksen nimi: Tietojen päivitys	
Toimijat:	Virkailija
Yleisyys:	Uuden asiakkaan tullessa järjestelmän piiriin tai jo olemassa olevan asiakkaan tietojen muuttuessa
Esiehdot:	Asiakkaan tiedot löytyvät järjestelmästä.
Kuvaus:	Kirjataan järjestelmään asiakkaan päivitettävät tiedot.
Lopputulos:	Asiakkaan tiedot järjestelmässä ovat ajan tasalla.
Poikkeukset:	P1. Tietokantaan ei saada yhteyttä.

Taulukko 1: Käyttötapaus tekstikuvauksella

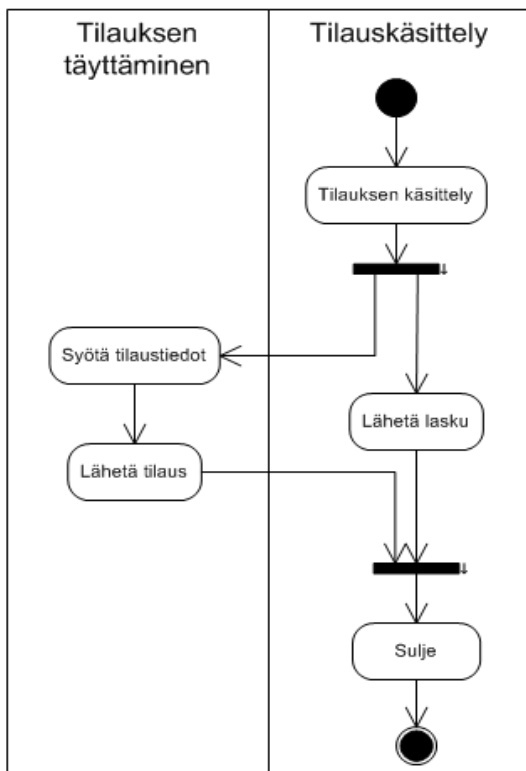


Kuva 8: Käyttötapauskaavio

4.3 Toimintokaavio

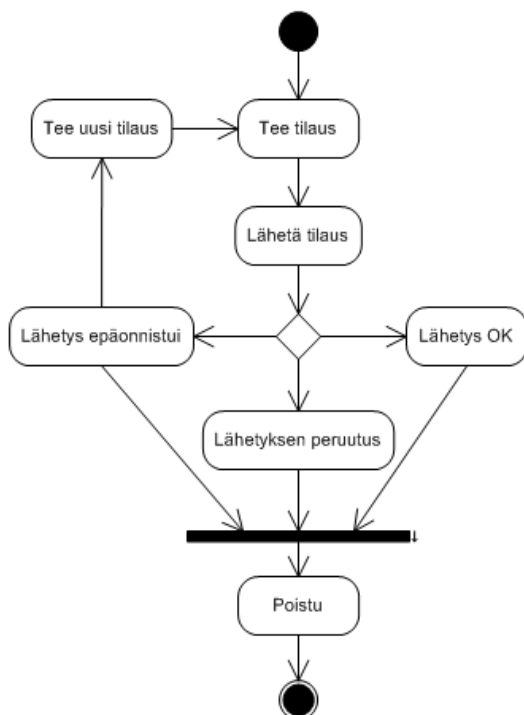
Toimintokaavioiden (activity diagrams) tarkoituksena on kuvata tehtävien tapahtumajärjestystä. Toimintokaaviot kuvaavat toimintoja ja niiden seurauksia. Toimintotila siirtyy seuraavaan tilaan heti, kun tilassa määritelty toiminto suoritettu. Toimintokaaviota voi käyttää yksittäiseen käyttötapaukseen liittyviä tapahtumia. Toimintokaaviolla voi kuvata käyttötapauksen skenaarion.

Toimintokaavioiden ominaisuutena on se, että ne voidaan sijoittaa uimaradoille (swimlanes). Uimaradoilla voidaan ryhmitellä toimintoja, niiden vastuualueen mukaan. Uimaradat kuvataan pystysuuntaisina kaiteina ja uimaradalle kuuluvat toiminnot sijoitetaan suorakaiteen sisään. Jokaiselle uimaradalle annetaan nimi, joka laitetaan suorakaiteen ylälaitaan. Kuvassa 9 on esimerkki toimintokaaviosta, jossa on kaksi uimarataa. Tilaus on ryhmitelty niin, että ensin vastaanotetaan tilaus, jonka jälkeen se jakautuu kahteen suoritettavaan toimintahaaraan (laskun lähettäminen ja tilauksen syöttäminen). Lopuksi nämä yhdistyvät ja rinnakkain suoritettavat toimintosarjat päättyvät.



Kuva 9: Toimintokaavio uimaradoilla

Toimintokaavioita voidaan käyttää eri käyttötarkoituksiin. Yleisin käyttötarkoitus on toimintojen kuvaaminen. Toimintokaavio näyttää toiminnat ja niiden väliset suhteet ja voi sisältää aloitus- ja lopetuspisteet. Kuvassa 10 on kuvattu toimintokaavio tilauksen tekemisestä. Jokaisen suorituksen jälkeen seuraa toinen toiminto, kunnes poistutaan toimintotilasta. (Fowler 2002, 113-123.)



Kuva 10: Toimintokaavio

5 Tietokanta ja tieto

Tänä päivänä tietokantoihin on lähes mahdotonta olla törmäämättä. Jokainen ihminen on jollakin tavoin tekemisissä tietokantojen kanssa. Tietokanta (Database, DB) on loogisesti yhteen kuuluvien, tallennettujen tietojen joukko, jota pystytään käsittelemään tietokantakielellä. Tiedolla tarkoitetaan tosielämän asioita, jotka voidaan kirjata ja joilla on jokin merkitys. Tietokannan käyttötarkoituksena on varastoida erilaista tietoa. Tietokantoihin voidaan päivittää, lisätä, muokata, hakea ja poistaa tietoa. Yritykset varastoivat tietoja asiakkaista, tilausmääristä ja tuotteista. SQL (Structured Query Language) -kielellä voidaan hakea tietoa esimerkiksi tietystä tuoteryhmästä tai tilausmääristä. Tietokantaohjelmat antavat

parhaan mahdollisuuden tiedon varastointiin, etsimiseen ja esittämiseen. (Hovi 2005, 4-5.)

Tietokannan ominaisuuksia:

- Tietokanta esittää jotain tosimaailman asioita
 - Tietokanta on loogisesti yhtenäinen kokoelma tietoa, jolla on jokin merkitys
- Tietokanta on suunniteltu, rakennettu ja täytetty tiedolla jotain tiettyä tarkoitusta varten. (Lahtonen 2002, 2-3.)

Tietokanta voi olla pieni ja yksinkertainen tai vastaavasti erittäin suuri ja monimutkainen. Se voi olla hyvinkin pieni ja yksinkertainen. Haku-, lisäys-, muokkaus- ja poisto ovat perustoimintoja tietokannan hallinnassa. Monimutkainen tietokanta on esimerkiksi verkkokauppa. Se sisältää paljon tietoa ja sen hallintaan tarvitaan enemmän tallennus tilaa sekä toimintoja. Tietokannalle kannattaa asettaa seuraavanlaisia vaatimuksia:

- Jokainen tieto tallennetaan tietokannassa yhteen paikkaan, jotta tietokannassa ei esiintyisi turhaa toistoa
- Tietokannan rakenteen muuttaminen pitäisi olla joustavaa
- Tietokannan käyttäminen ja sovellusohjelmat eivät ole riippuvaisia fyysisestä tallennus rakenteesta
- Tietojen etsiminen voidaan tehdä joustavasti erilaisin perustein ja myös sellaisin, joita ei tietokantaa suunniteltaessa ole pystytty ennakoimaan. (Lahtonen 2002, 2-3.)

5.1 Relaatietietokanta

Relaatietietokanta (relational database management system, RDBMS) on tietokantamalli. Relaatiomalli on SQL-pohjainen (relational database, RDB). SQL-kielen avulla voidaan muokata tietokannan rakennetta, kysellä tietokannan tietoja tai päivittää tietokannan tietoja. Relatiokannoilla voidaan toteuttaa operatiivisia sovelluksia ja tietovarastoja.

Tietomalli on kuvausmenetelmä, joka sisältää joukon erilaisia sääntöjä. Tietomallin avulla voidaan kuvata tietorakenteita ja niiden välisiä yhteyksiä. Relaatiomallin lisäksi muita tietomalleja ovat olio- ja käsittemallit. Relaatiomalleissa käytetään seuraavia termejä, kuten relaatio, attribuutti, yhteydet, viite- ja perusavain.

Relaatiomallin suunnittelu on viisasta jakaa kolmeen osaan:

- rakenne
- käsittely
- eheysäännöt. (Hovi 2005, 6-8.)

5.2 Relaatiotietokannan peruskäsitteet

Relaatiomallin on esittänyt E.F. Codd vuonna 1970. Relaatiotietokannan tiedot esitetään tauluina, joita kutsutaan myös relaatioiksi. Taulu (table) on joukko yhteen aiheeseen liittyviä tietoja. Esimerkiksi työntekijärekisterissä on omat taulunsa työntekijät, osasto, projekti jne. Relaatiomallin etuina ovat seuraavat asiat:

- Tiedon nopea siirto
- Tietokanta rakenteen helppo muuttaminen
- Tiedon esittäminen loogisesti, käyttäjillä oltava tietämys datan tallennuksesta.

Taulukon vaakasuorasta rivistä käytetään nimitystä tietue (record). Tietue sisältää tieto yhdestä kohteesta, esimerkiksi yhdestä työntekijästä tai yhdestä osastosta. Taulukon jokainen pystysuora sarake on nimeltään kenttä (field). Kenttä sisältää aina yhdenlaista tietoa, esimerkiksi yrityksen työntekijöiden etunimi, sukunimi ja työntekijän tunnus.

Kentän ja tietueen leikkauskohta on solu. Soluun tallennetaan yksi tiedon arvo. Solussa voi olla myös ns. tyhjäarvo. Sitä voidaan käyttää, kun tiedon arvoa ei vielä tiedetä tai arvo on mahdoton. Taulukossa henkilöt voisi yhtenä kenttänä olla synnytettyjen lasten lukumäärä. Naisilla lukumäärä on nolla tai positiivinen lukumäärä, miehillä voidaan käyttää tyhjää arvoa osoittamaan synnyttämisen

mahdottomuutta. Kenttiä luotaessa määritellään niille yleensä seuraavanlaisia ominaisuuksia:

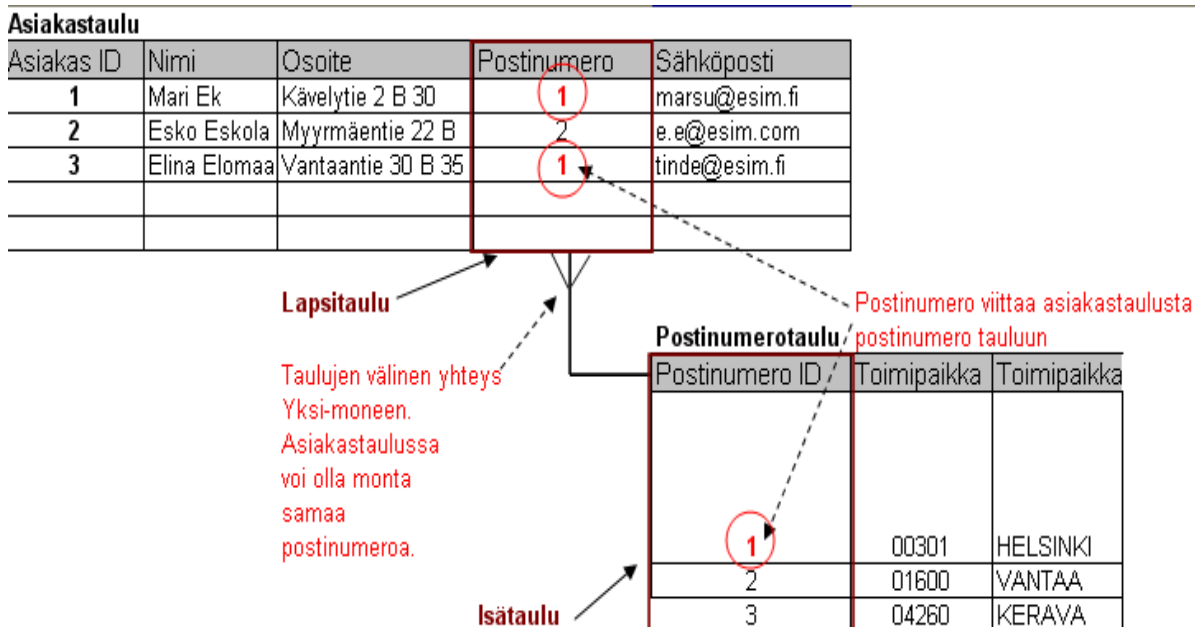
- Kentän nimi
- Tietotyyppi, joka voi olla numero, teksti, päivämäärä ja/tai kelloaika, looginen tai bittijono
- Kentän pituus ja mahdollinen tarkkuus
- Onko tieto pakollinen
- Tarkistukset syötettävän tiedon oikeellisuudelle tai sallittujen arvojoukkojen joukko
- Oletusarvo, jota käytetään jätettäessä kenttä tietoja syötettäessä tyhjäksi
- Syöttömaski, jolla kuvataan kenttään kelpuutettavat merkit ja niiden muoto.

Kentän nimeksi kannattaa valita jokin selkeä ja kuvaava nimi. Nimensä kannattaa kuitenkin välttää välilyöntejä, erikoismerkkejä ja skandinaavisia kirjaimia (å, ä, ö). Tietotyypit ja niiden nimet vaihtelevat hieman eri tietokantaohjelmistojen välillä. Kentän maksimipituutta määriteltäessä on oltava erityisen huolellinen. Määriteltäessä kentän pituutta tyypillisin virhe on varata kentälle liian vähän tilaa. Lähes kannattaa varata kentälle enemmän reilusti tilaa kuin liian vähän.

Tietokantaa suunniteltaessa pitää miettiä valitaanko kiinteämittainen vai vaihtuvainen kenttä. Kiinteämittaisessa kentässä varataan aina saman verran tilaa riippumatta siitä, kuinka paljon siihen tallennettava tieto oikeasti tarvitsee. Vaihtuvaisessa taas tilanvaraus muuttuu tarpeen mukaan kuitenkin ylittämättä määriteltyä maksimipituutta.

Mahdollisimman moni kenttä kannattaa määritellä pakolliseksi tai käyttää oletusarvoa. Tällöin vältetään tyhjäarvo-ongelmia. Null on erityinen merkintätapa puuttuvalle arvolle. Null ei tarkoita välilyöntiä tai nolaa. Jos kentällä ei syöttövaiheessa anneta mitään arvoa, niin sen sisällöstä tulee null. Se tarkoittaa lähinnä tuntematonta arvoa eli sen arvo voisi periaatteessa olla mikä tahansa tai ei mikään. Kuvassa 11 on esimerkki relaatiomallin taulusta ja sen osista. (Lahtonen 2002, 4-8.)

viiteavaimen-syntaksilla. Lisäksi kummallakin taululla pitää olla sama arvojoukko. Viiteavain muodostaa siis yhteyden isä- ja lapsitaulun välille (ks. kuva 12).



Kuva 12: Taulujen välinen yhteys ja viittaus

Mikäli isä-taulusta halutaan poistaa tai muuttaa arvoja on otettava huomioon onko viite-eheys rikkoutumassa. Jos isä-taulussa olevia arvoja on lisättävä, muutettava tai poistettava on olemassa kolme peruseriaatetta, minkä mukaan muutoksia voidaan tehdä ja nämä toimintosäännöt ovat seuraavat:

- Viiteavaimen nollaus (nullify)
- Oletusarvon asettaminen (set default)
- Johdannaismuutos (cascade)
- Rajoitettu muutos (restricted).

(Lahtonen 2000, 12-13.)

6 Käytettävyys ja käyttöliittymän suunnittelu

Käytettävyydellä tarkoitetaan menetelmää- ja teoriakenttää, jonka kautta käyttäjän ja laitteen yhteistoimintaa pyritään saamaan tehokkaammaksi ja käyttäjän kannalta miellyttävämmäksi. (Sinkkonen ym. 2006, 17).

Viime aikoina käytettävyys, käyttökokemus ja käyttäjäystävällisyys ovat herättäneet paljon keskustelua. Käyttöliittymän suunnittelussa voidaan käyttää seuraavaista yksinkertaista mallia, jonka on kehittänyt käytettävyysasiantuntija Keith Instone. Hän on luonut yksinkertaisen heuristisen mallin. Tämän yksinkertaisen tavan ideana on selvittää käyttöliittymän käyttäjän perustarpeita ja odotuksia. Ideana on vastata kysymyksiin:

1. Missä minä olen?
2. Mitä täällä on?
3. Minne täältä pääsee?

Mallia voi kokeilla vaikka heti avaamalla minkä tahansa Web-sivun. Selviääkö sivulta minne käyttäjä on saapunut? Käykö sivun tarkoitus ilmi yhdellä silmäyksellä? Onko sivuston yleinen rakenne helppo hahmottaa ja onko käyttäjän helppo päätellä minne sivulta pääsee? (Veen 2002, 47-48.) Käytettävyys ja käyttöliittymän suunnittelu on otettava huomioon kanta-asiakassovelluksen määrittelyvaiheessa.

Nielsenin mukaan käytettävyys on tärkeä asia, mikä tarkoittaa sitä, että käyttäjä on kuningas ja päättää kaikesta. Jos palvelu ei tyydytä, ei asiakas sitä myöskään halua. Käyttäjän huomio kiinnittyy sivujen ulkoasuun ja näkee vain yhden sivun kerrallaan. Käyttäjän liikkuminen tulee ottaa huomioon suunniteltaessa käyttöliittymää ja sitä pitää tukea (Nielsen. 2000, 9, 25.). Hyvän käytettävyyden muodostavat käyttötilanteessa opittavuus, virheettömyys, muistettavuus, tehokkuus ja miellyttävyys. (Sinkkonen ym. 2006, 17).

Kanta-asiakassovelluksen käyttöliittymän suunnittelussa on otettava huomioon monia eri asioita: silmäiltävyys, vasteaika, värit, sommittelu ja paloittelu, liikkuminen, navigointi, metaforat, sivujen koko, otsikointi (nimeäminen), linkkien värit, tekstin koko sivulla (ja luettavuus) ja kuvat.

Virheilmoitukset tai poikkeukset kannattaa suunnitella käyttäjälle ymmärrettävään muotoon. Virheilmoitus ei saa olla sovelluksen kielellä. Esimerkiksi huonosta virheilmoituksesta on: Error # 35 ”Tyhjä arvo”. Tällainen virheilmoitus ei ole käyttäjän kannalta viisas. Hyvä virheilmoitus esimerkki on: ”Lomakkeella on virheellinen henkilötunnus. Ole hyvä ja tarkista tieto. Tiedon muoto on pitää olla

xxxxxx-xxxx mallin mukainen”. Käyttäjän on helpompi korjata tai käsittää virheilmoituksen merkitys. (Sommerville 2001, 341.)

7 Kanta-asiakassovelluksen määrittelyprosessi

Kanta-asiakassovelluksen määrittelyprosessi oli pienimuotoinen projekti, missä ei ollut kiireellistä aikataulua, ei kustannuksia tai budjettia. Projekti oli kertaluonteinen työ, mikä alkoi kesällä 2009 ja se päättyi toukokuussa 2010, kun haluttu lopputulos saavutettiin. Osallistujina projektissa olivat tutkimuksen tekijä Mari Ek ja toimeksiantajana Formlek Oy:n omistaja Seppo Weckman. Työn ohjaajana toimi Laurea-ammattikorkeakoulun lehtori Juha Uutela. Tämä ryhmä muodosti ns. projektiryhmän. Mari Ek toimi tutkijana raportoiden työn etenemisestä Seppo Weckmanille ja Juha Uutela ohjasi työtä.

Toiminnallinen määrittelydokumentti ja tutkimusraportti tuotettiin erillisinä dokumentteina. Syy tähän oli yksinkertainen. Kanta-asiakassovelluksen määrittelydokumentti on tehty suoraan toimeksiantajan tarpeisiin ja käyttöön. Tutkimusraportti on tarkoitettu luettavaksi tietojenkäsittelyalan opiskelijoille ja opettajille tai samasta aiheesta tutkimusta tekeville. Aluksi tutkimusta ei suunniteltu tehtäväksi erillisinä dokumentteina, se vain tuntui loogiselta vaihtoehdolta.

7.1 Kanta-asiakassovelluksen esitutkimus ja vaatimuskartoitus

Tutkimus aloitettiin ongelman määrittelyllä, tehtävän tarkentamisella ja aiheen rajaamisella. Esitutkimuksen avulla kartoitettiin toimeksiantajan asiakasvaatimuksia sovelluksesta. Tapaamisissa vaatimuksia kirjoitettiin paperille muistiin, jotta idean hahmottaminen olisi helpompaa. Lähtökohtana oli se, että Formlek Oy on kehittänyt kanta-asiakasformaatin, mikä pitäisi muuntaa toiminnallisen määrittely muotoon. Tarkennettavia kysymyksiä lähetettiin myös sähköpostin välityksellä. Esitutkimuksen avulla selvitettiin miksi kanta-asiakassovellus kehitetään, onko sellainen mahdollista tehdä, millaista tietoa se tulee sisältämään ja millaisia toimintoja siinä pitäisi olla, millaiseen ympäristöön sitä ollaan kehittämässä, miten se aiotaan toteuttaa?

Esitutkimuksen jälkeen käytiin läpi kehitettävän kanta-asiakassovelluksen vaatimuksia. Vaatimukset olivat seuraavanlaisia:

- Pitää toimia Web - selaimella (kanta-asiakkaan käyttöliittymän hallinta)
- Sovellus kehitetään markkinoinnin helpottamiseksi
- Markkinoinnin kohdistaminen
- Tarjousten hallinta (tarjouspyyntöjen ja tarjouksien lähettäminen)
- Heräteostoksien lisääminen
- Vaihtoehto puhelinmyynnille
- Asiakastietojen varastointi (asiakasrekisteri)
- Kanta-asiakkaaksi pääseminen vaatii rekisteröitymisen
- Sovelluksen pitää olla turvallinen, esim. tunnustautuminen
- Asiakas voi valita/valtuuttaa markkinointitavan myyjälle
- Asiakkaalla on käytössä profiili (profiilissa määritellään markkinointitavat ja kiinnostuksen kohteet)
- Asiakas pystyy rakennehaun avulla etsimään tarjouksia
- Asiakas voi jättää yhteydenottopyynnön liikkeessä kanta-asiakastunnuksella
- Markkinointikeinot: posti, tekstiviesti, sähköinen mainos (kanta-asiakassivulle, HTML -pohjainen), sähköposti ja soitto
- Sovelluksessa on pystyttävä käyttäjän päivittämään tietoja tarpeen mukaan (perustiedot, profiilitiedot)
- Sovelluksen kautta pääsy verkkokauppaan
- Sovelluksen pitää pystyä ottamaan vastaan/hallitsemaan asiakkaan yhteystietoja
- Tuotteiden esittely ja myynti
- Loki virhetilanteiden analysointiin.

7.2 Toiminnallisen määrittelydokumentin luominen

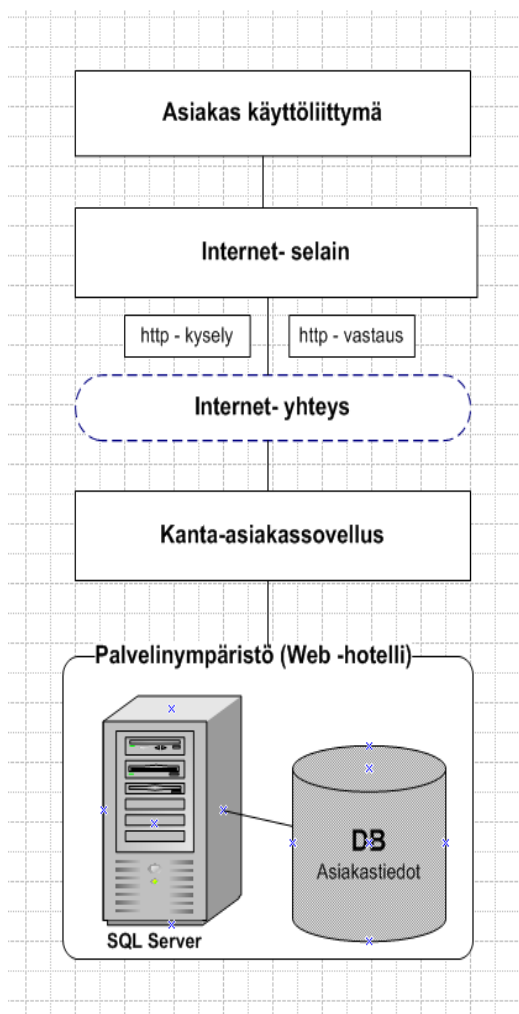
Esitutkimuksen ja vaatimuskartoituksen perusteella aloitettiin toiminnallisen määrittelydokumentin luominen. Dokumentointimalleja etsittiin kirjallisuudesta ja WWW-sivuilta. Toiminnallinen määrittelydokumentti lähetettiin aina tarpeen mukaan tarkastettavaksi toimeksiantajalle. Tarkastuksen jälkeen dokumenttiin tehtiin tarvittavia korjauksia. Tällä tavalla pyrittiin pitämään dokumentin laatu ja raportointi ajan tasalla.

Kanta-asiakassovelluksen toiminnallinen määrittelyn ensimmäinen dokumenttimalli luotiin marraskuun lopulla 2009. Se sisälsi perustietoja mm. kansilehden,

sisällysluettelon, alustavan otsikoinnin sisällöstä ja versionhallinnan. Vastatammikuussa 2010 dokumentti muutettiin IEEE 830-sisältörungon mukaiseksi. Dokumentointimalli otettiin suoraan mallina Hirsjärvi & Märijärven kirjasta. Tietojen lisäämistä dokumenttiin tehtiin tasaisesti kevään aikana. Versiohallintaan kirjattiin ainoastaan oleelliset ja merkittävät muutokset sekä lisäykset. Toiminnallisen määrittelyn tuloksena syntyi kanta-asiakassovelluksesta määrittelydokumentti, mikä perustuu vaatimuksien kartoitukseen ja esitutkimukseen. Kanta-asiakassovelluksen toiminnallisen määrittelydokumentin pituudeksi tuli 36 sivua. Dokumentin kappaleet 1-4 kuvaavat toiminnallisia vaatimuksia ja kappaleet 5-6 ei-toiminnallisia vaatimuksia. Seuraavaksi kerrotaan lyhyesti dokumentin luomisen vaiheita ja sen sisällöstä. Toiminnallinen määrittelydokumentti on kokonaisuudessaan liitteessä 1.

Luku yksi sisältää johdannon, missä kerrotaan kanta-asiakassovellusideasta, dokumentin tarkoituksesta, tuotteesta ja selvennetään sovellukseen liittyviä teknisiä ja sovellukseen liittyviä määritelmiä mm. kanta-asiakas, myyjä, sallittu markkinointitapa, tarjous, profiili, tarjouspyyntö, kategoria, sähköinen mainos ja virtuaalinen kanta-asiakaskortti.

Luvussa kaksi kuvattiin yleisellä tasolla kanta-asiakassovelluksen ympäristöä, toimintaa, käyttäjiä ja rajoitteita. Toimintaympäristö (ks. kuva 13) koostuu asiakkaan käyttöliittymästä, selaimesta, tietoliikenne yhteydestä, sovelluksesta, palvelinympäristöstä ja tietokannasta. Tiedonsiirto tapahtuu haulla, kutsuilla, kyselyillä ja vastauksilla.



Kuva 13: Toimintaympäristö

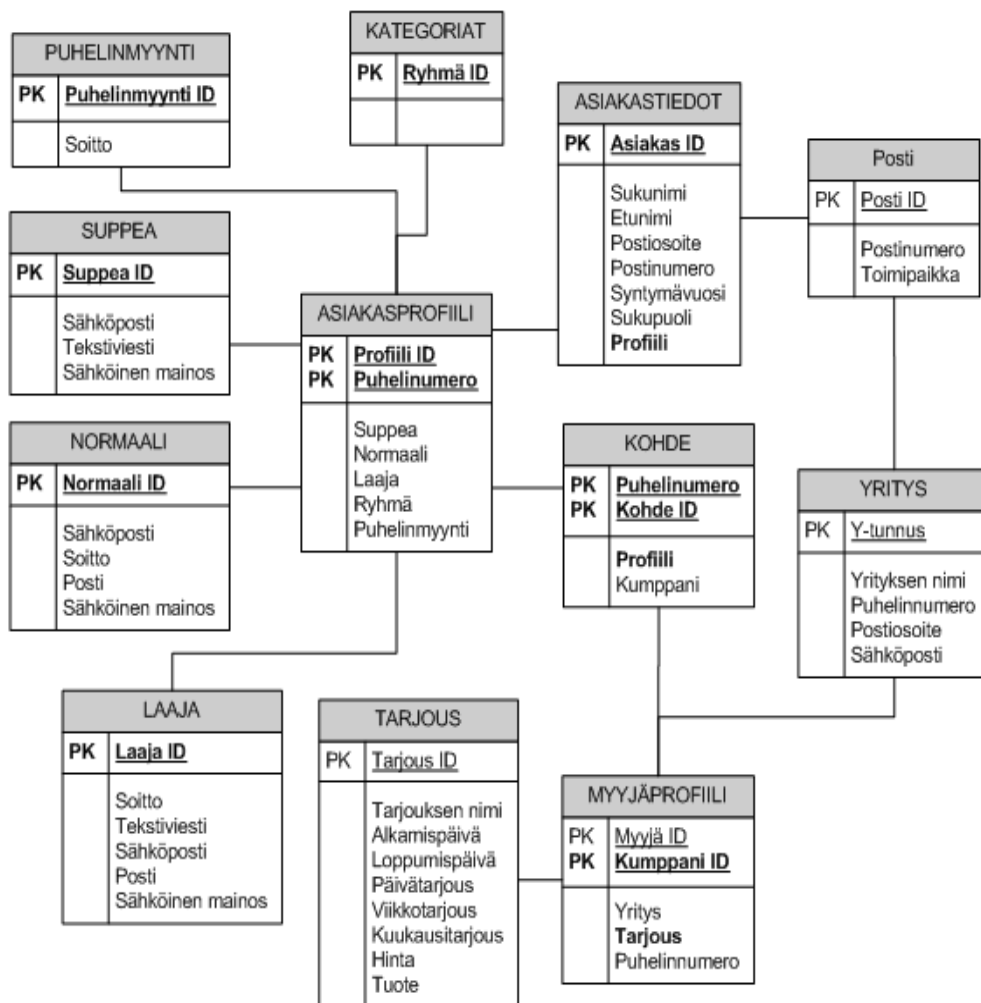
Sovelluksen käyttäjät ovat asiakas, myyjä ja ylläpitäjä. Käyttäjä rooleja kuvattiin käyttötapauksia hyödyntäen. Käyttötapauksessa kuvattiin käyttäjän toimintaa ympäristössä ja käyttäjävaatimuksia. Käyttötapaukset tehtiin taulukoilla, mikä sisälsi seuraavat tiedot: Käyttötapauksen (nimi), käyttäjä(t), esiehdot, kuvauksen, poikkeukset ja lopputuloksen. Taulukossa 2 on esimerkki yhdestä dokumentin käyttötapauksesta.

Käyttötapaus 2: Tietojen päivittäminen: muokkaus, poisto, valinta, lisäys	
Käyttäjät:	Asiakas
Esiehdot:	Asiakastietojen pitää olla järjestelmässä. Asiakkaan pitää olla

	kirjautuneena omalla hallintasivulla.
Kuvaus:	<ol style="list-style-type: none"> 1. Asiakas selaa omia tietoja. 2. Asiakas haluaa päivittää tiettyjä asiakas- tai profiilitietoja ja tekee sen jälkeen tallennuksen. 3. Muutokset menevät suoraan tietokantaan ja järjestelmä ilmoittaa onnistuneesta tietojen päivittämisestä.
Poikkeukset:	<p>P1. Tietojen päivittäminen ei onnistunut. Pakolliset kentät pitää olla täytettynä ja tieto oikeassa muodossa.</p> <p>P2. Yhteys tietokantaan on poikki.</p>
Lopputulokset:	Tietojen muutokset onnistuivat ja tallentuivat tietokantaan onnistuneesti.

Taulukko 2: Käyttötapa tietojen päivittämisestä

Tietokannan rakenne kuvattiin relaatiomallilla (ks. kuva 14). Tiedon tarkemmasta sisällöstä tehtiin erillinen Excel-tilasto. Tiedoista kuvattiin taulunimi, kentännimi, tietotyyppi ja sen pituus, onko pää- tai viiteavainta ja kuvaus. Tietojen tarkemman kuvauksen voi katsoa liitteessä 1 ja kappaleesta 3.2.

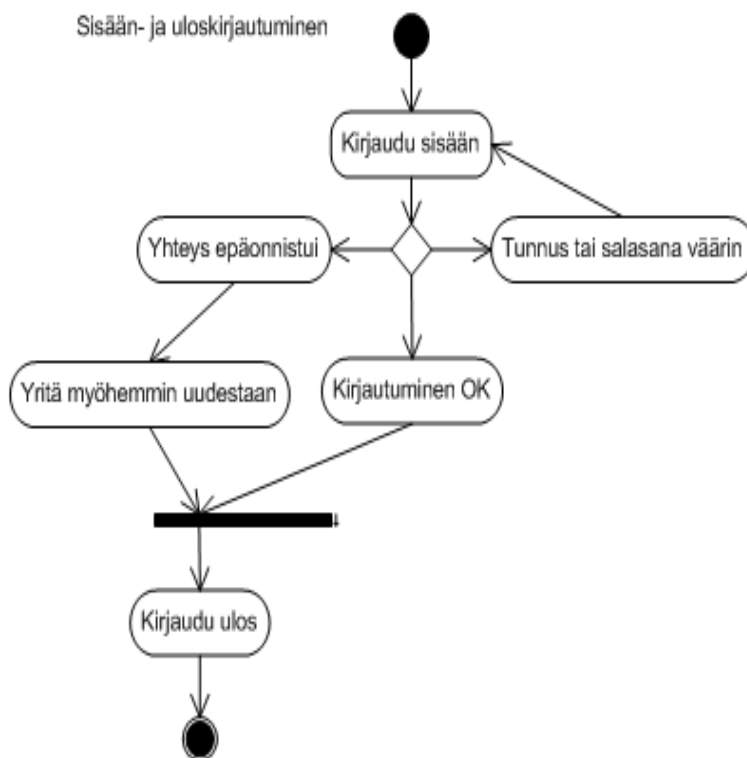


Kuva 14: Kanta-asiakassovelluksen tietokanta

Toimintojen kuvaukset suunniteltiin alun perin tehtäväksi käyttötapa-kaavioilla, mutta huomattiin tietomallikaavion olevan tähän parempi vaihtoehto. Dokumentissa kuvattiin seuraavat toiminnot: tarjouksen lähettäminen, kirjautuminen sisään ja ulos sovelluksesta, rekisteröityminen, tietojen päivittäminen, tarjouspyynnön ja tarjouksen lähettäminen. Dokumentissa kuvatusta kaaviosta on esimerkki kuvassa 15. Kaaviossa kuvataan toiminto sisään ja uloskirjautuminen. Kaavioiden lisäksi kirjoitettiin tekstipohjainen selitys kuvien tueksi.

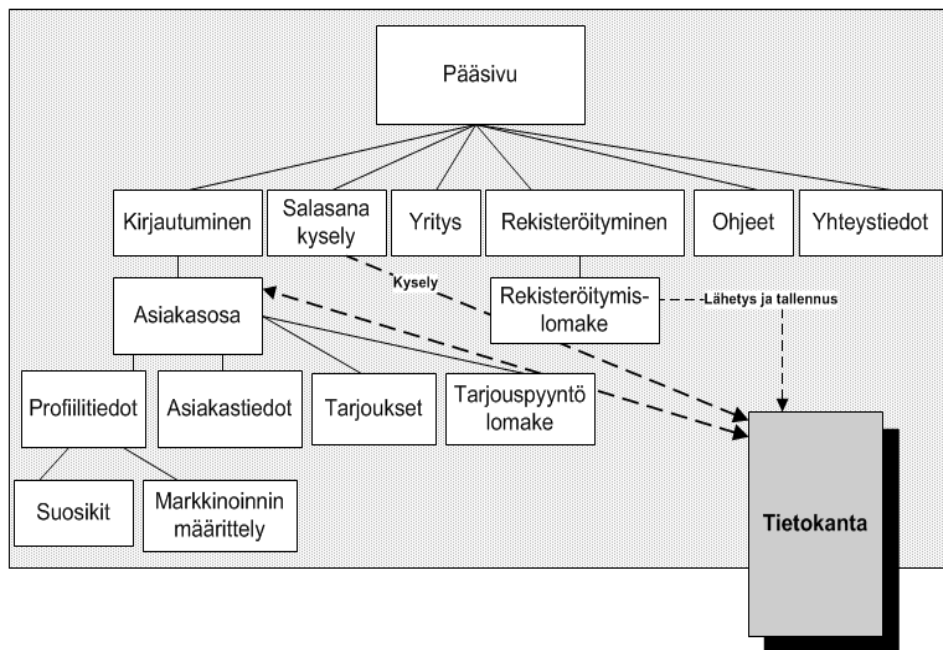
Toiminto:

Sisään- ja uloskirjautuminen




Kuva 15: Tietomallikaavio kirjautumisesta sisään ja ulos

Käyttöliittymän alustava määrittely kuvattiin luvussa viisi kappaleessa 5.1. Käyttöliittymästä kuvattiin sovelluksen-sivurakenne (ks. kuva 16), protomalli ja erilaisia komponentteja. Komponentit kuvattiin erillisessä Excel-taulukossa, missä esitetään: nimi, osa, toiminto, tapahtuma, mitä seuraa, jos toiminto ok tai toiminnossa tulee virhe ja kuva mahdollisesta komponentista. Käyttöliittymän ulkonäöstä tehtiin protomalli (ks. kuva 17). Protomalli luotiin Vision-työkalulla ja tekijän luovuutta hyödyntäen. Protomallissa kuvataan kirjautumissivua. Toinen protomalli esimerkki on rekisteröitymislomakkeesta kuvassa 18.



Kuva 16: Sivuston rakenne

Valtuutettu Markkinoija

Sulje 

Valtuutetun markkinoijan avulla voit hallita markkinointitapoja ja tehdä tarjouspyyntöjä tuotteista sekä vastaanottaa tarjouksia.

Kirjaudu sovellukseen:

Minulla ei ole tunnuksia? [Rekisteröidy tästä.](#)

Salasana unohtunut? [Uusi salasana tästä.](#)

Formlek Oy © 2010 [Yritys](#) [Yhteystiedot](#) [Ohjeita](#) [Käyttäjähdot](#)

Kuva 17: Protomalli käyttöliittymästä

Täytä rekisteröintilomake ja lähetä lomake:

Pakolliset tiedot on merkitty (*) :llä.

(*)
 (*)
 (*)
 (*)
 (*)
 (*)
 Valitse itsellesi sopiva käyttätunnus ja salasana:
 (*)
 (*)
 (*)

Kuva 18: Protomalli rekisteröitymislomakkeesta

Luvussa viisi kerrotaan käyttöliittymän lisäksi seuraavista asioista: laitteisto- ja ohjelmistoliittymä, turvallisuus ja suojaus, ylläpidettävyys, operointi, siirrettävyys ja yhteensopivuus.

Viimeisessä kappaleessa kuvattiin suunnittelu rajoitteet: standardit, laitteisto- ja ohjelmistorajoitteet. Lyhyesti voidaan kertoa, että sovellus tulee toimimaan Web -selaimen kautta, joten mahdollisimman uutta selainta suositellaan. Standardit kehitystyössä tulevat olemaan: C#, AJAX, ASP.NET, SQL. Dokumentissa kartoitettiin myös mahdolliset laitteistorajoitteet:

- Tietokanta: MsSQL
- Palvelin: Windows 7
- Web-hotelli
- Microsoft-kehitysympäristö, Visual Studio 2008 tai 2010.

Dokumentin lopussa on lähde-, kuva- ja taulukkoluetelot.

8 Yhteenveto

Opinnäytetyö sai alkunsa siitä, kun lehtori Heikki Lempola ehdotti aihetta ja esitteli toimeksiantajan. Ohjelmistotuotannon laajuudesta johtuen aihetta oli aluksi vaikea rajata. Lopulta päädyttiin siihen tulokseen, että keskitytään ainoastaan määrittelyprosessin tuottamiseen. Tutkimuskysymyksen avulla etsittiin vastausta kanta-asiakassovellusformaatin muuntamisesta toiminnalliseksi määrittelyksi määrittelyprosessin mukaan. Aihe oli tutkijalle vierasta aluetta, minkä takia tietämys ei ollut parhaalla mahdollisella tasolla. Aluksi aikaa kului kirjallisuuteen tutustumiseen ja aiheen sisäistämiseen.

Ohjelmistoprojektit ovat aikaa vieviä ja vaativia prosesseja. Sen takia on syytä panostaa määrittelyyn ja dokumentoimiseen. Ohjelman korjaaminen määrittelyvaiheessa on halvempaa kuin käyttöönottovaiheessa. Dokumentoinnin tekeminen oli ajoittain työlästä ja ohjelmistotuotannon ymmärtäminen haasteellista. Määrittelyprosessin merkitys on ohjelmistotuotannossa merkittävä, koska siinä selvitetään sovellukselle asetettuja asiakasvaatimuksia. Ilman oikeita vaatimuksia kehitettävän sovelluksen toiminnot voivat olla vääriä tai puutteellisia.

Tutkimuksen aikana käytetyt menetelmät olivat toimia ja niiden hyödyntäminen onnistui hyvin. Microsoft Vision-ohjelmalla oli helppo tehdä erilaisia kaavioita. RUP-menetelmän käyttäminen olisi voinut olla toimivampi vaihtoehto määrittelyprosessissa, koska sen avulla voidaan hallita kokonaisuudessa ohjelmistokehitystä ja siihen on kehitetty parhaiten soveltuvat työkalut. RUP-prosessia ei valittu menetelmäksi sen takia, että tekijän työympäristössä ei tätä prosessityökalua ollut mahdollista käyttää. Toinen vaihtoehto olisi ollut käyttää Microsoft Visual Studio-ohjelmistoa. Visioin käyttämiseen päädyttiin sen takia, että siitä oli jo ennestään käyttökokemusta.

Tutkimuksen tavoitteet saavutettiin ja lopputuloksena syntyi toiminnallinen määrittelydokumentti Formlek Oy:lle. Asiakasvaatimukset muunnettiin toiminnallisen määrittelyksi. Toimeksiantaja hyväksyi määrittelydokumentin 28.4.2010 viimeisessä tarkastuksessa, minkä jälkeen se versiointiin numeroksi yksi. Tuloksen syntymisen vaiheet kerrottiin tutkimuksessa pääpiirteittäin ja kokonaisuudessaan lopputulosta voidaan tarkastella liitteestä 1. Kokonaisuudessaan tutkimuksen määrittelyprojekti onnistui hyvin. Toimeksiantaja oli tyytyväinen tulokseen ja antoi tehdystä työstä

positiivista palautetta. Palautteessa kerrottiin työn sisällön, työskentelytavan ja lopputuloksen olleen ammattimaista tasoa. Työn tekijän henkilökohtainen osaaminen, oppiminen ja ymmärrys ohjelmistokehityksestä kehittyivät tutkimusprosessin aikana. Määrittelyprosessin sisäistäminen tuntui aluksi hankalalta, mutta kirjallisuutta lukiessa se hahmottui paremmin. Käsitys ohjelmistokehityksestä luonteesta avautui ja syvensi tietoutta.

Jatkotyöehdotuksena aiheesta voisi jatkaa tutkimusta esimerkiksi kanta-asiakassovelluksen suunnitellulla (tekninen määrittely) tai kanta-asiakassovelluksen toteutusvaiheessa testauksella.

Lähteet

Fowler, M. & Scott, K. 2002. UML. Jyväskylä: Docendo Finland.

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. Helsinki: Satku.

Hovi, A., Huotari, J. & Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. Porvoo: WS Bookwell.

Kuoppala, P., Parkkinen, J., Sinkkonen, I. & Vastamäki, R. 2006. Käytettävyyden psykologia. Helsinki: Edita.

Järvinen, A. & Järvinen, P. 2004. Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Lahtonen, T. 2002. SQL. 2. painos. Jyväskylä: Docendo.

Lukka, K. 2001. Konstruktiivinen tutkimus.
http://www.metodix.com/fi/sisallys/01_menetelmat/. Viitattu 30.11.2009.

MainosAvain. 2010. Yritys.
<http://www.mainosavain.fi/Yritystiedot.aspx>. Viitattu 20.4.2010.

McConnell, S., 2002. Ohjelmistotuotannon hallinta. Helsinki: Edita.

Metropolia. Toimintokaavio.
http://users.metropolia.fi/~k0201257/koulu/uml/luennot/UML_toimintokaavio.pdf. Viitattu 20.4.2010.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo.

Seppänen, V. 2004. Konstruktiivinen tutkimus.
http://74.125.77.132/search?q=cache:uElh4l8QUUkJ:media.tol.oulu.fi/video/jtmk/konstruktiivinen_tutkimus.ppt+konstruktiivinen+tutkimus&cd=1&hl=fi&ct=clnk&gl=fi&client=firefox-a. Viitattu 20.4.2010.

Sommerville, I. 2001. Software Engineering. Harlow: Addison-Wesley.

Veem, J. 2002. Inside Web design. Helsinki: Edita.

Kuvat ja kuviot

Kuva 1: Tutkimustyön rajaus	8
Kuva 2: Konstruktiivisen tutkimusotteen keskeiset elementit.....	10
Kuva 3: Toteutusprosessi	11
Kuva 4: Perinteinen vesiputousmalli	13
Kuva 5: Määrittelyprosessi	15
Kuva 6: Tuotantoprosessi.....	16
Kuva 7: Toiminnallisen määrittelyn sisältörunko-IEEE 830	19
Kuva 8: Käyttötapauskaavio	23
Kuva 9: Toimintokaavio uimaradoilla.....	24
Kuva 10: Toimintokaavio	25
Kuva 11: Relaatiomalli.....	29
Kuva 12: Taulujen välinen yhteys ja viittaus.....	30
Kuva 13: Toimintaympäristö	35
Kuva 14: Kanta-asiakassovelluksen tietokanta	37
Kuva 15: Tietomallikaavio kirjautumisesta sisään ja ulos	38
Kuva 16: Sivuston rakenne	39
Kuva 17: Protomalli käyttöliittymästä	39
Kuva 18: Protomalli rekisteröitymislomakkeesta	40

Taulukot

Taulukko 1: Käyttötapaus tekstikuvauksella	23
Taulukko 2: Käyttötapaus tietojen päivittämisestä	36

Liitteet

Liite 1: Kanta-asiakassovelluksen toiminnallinen määrittely **Error! Bookmark not defined.**

FORMLEK OY /MainosAvain -formaatti

Kanta-asiakassovelluksen toiminnallinen määrittely

1.0

Tarkastettu

Helsinki, 2010

Formlek Oy

Toiminnallinen määrittelydokumentti
Versio 1.0

Versionhallinta 1.0

Päivä	Versio	Kuvaus muutoksesta	Tekijä
21.11.2009	0.1	Dokumentin luominen	M. Ek
9.1.2010	0.2	Dokumentin muuttaminen IEEE830- standardin mukaiseksi.	M. Ek
10.1.2010	0.3	Tietojen lisääminen.	M. Ek
10.1.2010	0.4	Tietokanta taulujen lisääminen dokumenttiin.	M. Ek
22.2.2010	0.6	Tietokanta kuvauksen korjaaminen ja täydentäminen. Versiohallinnan korjaus.	M. Ek
26.2.2010	0.6.1	Tietokanta tietojen päivittäminen ja lisääminen	M. Ek
5.3.2010	0.7	Käyttötapauksien luominen ja kirjoittaminen	M. Ek
7.3.2010	0.7.1	Käyttötapauskaavion lisääminen	M. Ek
3.4.2010	0.7.2	Käyttötapauksien muokkaus	M. Ek
17.4.2010	0.8	Käsitteiden, lyhenteiden korjaaminen	M. Ek
17.4.2010	0.8.1	Tietokannan kuvauksen tekeminen loppuun	M. Ek
17.4.2010	0.8.2	Käyttötapauksien muokkaus	M. Ek
18.4.2010	0.8.3	Päivitetty koko dokumenttia	M. Ek
20.4.2010	0.9	Dokumentin muokkaamista ja viilaamista	M. Ek
24.4.2010	0.9.1	Käyttötapauksien siirtäminen 2 kappaleeseen ja Toimintokaavioiden lisääminen toiminnot osaan.	M. Ek
26.4.2010	0.9.2	Käyttöliittymä komponentti tietojen lisääminen. Käyttöliittymä osioon korjaus komponentti kuvaan.	M. Ek
28.4.2010	1.0	Toimintokaavioihin tekstiselitykset. Toiminnallinen määrittelydokumentti valmis.	M. Ek

Formlek Oy

Toiminnallinen määrittelydokumentti
Versio 1.0

Sisällys

1	Johdanto	5
1.1	Dokumentin tarkoitus	5
1.2	Tuote	5
1.3	Määritelmät, lyhenteet ja termit.....	5
1.4	Viitteet ja muut liittyvät dokumentit	8
1.5	Yleiskatsaus dokumenttiin.....	8
2	Yleiskuvaus	9
2.1	Ympäristö.....	9
2.2	Toiminta.....	10
2.3	Käyttäjät	11
2.4	Yleiset rajoitteet.....	16
2.5	Oletukset ja riippuvuudet	16
3	Tiedot ja tietokanta	17
3.1	Tietokannan rakenne ja taulujen väliset yhteydet	17
3.2	Tietokannan tarkempi kuvaus.....	18
4	Kanta-asiakassovelluksen toiminnot	19
4.1	Sisään ja uloskirjautuminen.....	20
4.2	Rekisteröityminen	21
4.3	Tietojen päivittäminen	23
4.4	Tarjouspyynnön ja tarjouksen lähettäminen.....	24
5	Ei-toiminannalliset vaatimukset	25
5.1	Käyttöliittymä	25
5.1.1	Kanta-asiakassovelluksen sivustorakenne.....	26
5.1.2	Alustavat käyttöliittymän komponentit.....	27
5.2	Laitteistoliittymät.....	30
5.3	Ohjelmistoliittymät.....	31
5.4	Turvallisuus ja suojaukset	31
5.5	Siirrettävyys ja yhteensopivuus.....	32
5.6	Ylläpidettävyys	32
5.7	Operointi (Käyttäjän ylläpitotoimet)	33
6	Suunnittelu rajoitteet.....	33

Formlek Oy

Toiminnallinen määrittelydokumentti
Versio 1.0

6.1	Standardit.....	33
6.2	Laitteistorajoitteet.....	33
6.3	Ohjelmistorajoitteet.....	34
	Lähteet	35
	Kuvat	36
	Taulukot.....	36

1 Johdanto

Formlek Oy on kehittänyt sovellusidean, josta on tarkoituksena kehittää kanta-asiakassovellus tuotteiden markkinointiin. Idea perustuu *MainosAvain* -formaattiin, minkä tekijänoikeudet omistavat Seppo Weckman.

Tämä määrittelydokumentti sisältää kanta-asiakassovelluksen toiminnallisen määrittelyn. Ensimmäisessä luvussa kerrotaan dokumentoinnin tarkoitus, mikä tuote kyseessä ja esitellään dokumentin keskeiset määritelmät, termit ja lyhenteet sekä dokumentin yleiskatsaus.

1.1 Dokumentin tarkoitus

Toiminnallisen määrittelydokumentin tarkoituksena on kuvata kanta-asiakassovelluksen kokonaisuutta, rakennetta ja sen tulevia toimintoja. Määrittelyn tarkoituksena on antaa tietoa sovelluksen toiminnallisista ja ei-toiminnallisista vaatimuksista. Dokumentin tarkoitus on helpottaa ohjelmiston suunnittelua ja toteuttamista sekä selkeyttää vaatimuksia.

1.2 Tuote

Sovellusta kutsutaan tällä hetkellä valtuutetuksi markkinoijaksi. Kanta-asiakassovellus on ohjelma, minkä avulla on tarkoitus helpottaa ja tehostaa markkinointia. Ideana on saada kohdistettua markkinointi, mainokset ja tarjoukset oikeille kohderyhmille ja helpottaa viestimistä. Sovelluksen avulla voidaan hallita asiakastietoja ja tarjouksia sekä ylläpitää asiakasrekisteriä. Sovelluksessa voidaan käyttää rakennehakua, jonka perusteella haetaan tarjouksia myyjien ylläpitämiltä sivuilta.

1.3 Määritelmät, lyhenteet ja termit

Sovellukseen liittyviä määritelmiä:

Kanta-asiakas:

Kanta-asiakas on kanta-asiakassovelluksen käyttäjä, joka etsii kiinnostavia ja edullisia tuotteita sovelluksen avulla. Asiakas voi tehdä tarjouspyyntöjä ja vastaanottaa tarjouksia myyjiltä. Asiakas hallitsee kanta-asiakassivuston kautta omia tietojaan, mm. henkilö- ja profilitietoja.

Myyjä:

Myyjä markkinoi ja myy tuotteita. Myyjät voivat ottaa yhteyttä asiakkaisiin sallituilla markkinointikeinoilla. Asiakasprofiilin avulla myyjät voivat kohdistaa oikeita tuotteita asiakkaalle.

Sallittu markkinointitapa:

Sallitulla markkinointitavalla tarkoitetaan sitä keinoa, miten myyjä voi markkinoida tuotetta asiakkaalle. Markkinointimuodot ovat valtion posti, sähköposti, soitto, sähköinen mainos tai tekstiviesti.

Tarjous:

Tarjous sisältää tarjouksen nimen, hinnan ja voimassaolo päivän. Tarjouksen voimassaolo ajat ovat mm. päivä, viikko tai kuukausi.

Profiili:

Asiakas pystyy omassa profiilissa muokkaamaan profilitietoja. Profilitiedoissa hallitaan sallittuja markkinointitapoja ja kiinnostuksen kohteita. Profiilia voidaan muokata milloin vain ja tiedot päivittyvät välittömästi tallentamisen jälkeen.

Tarjouspyyntö:

Asiakas voi jättää myyjälle tarjouspyynnön liikkeessä virtuaalisella kanta-asiakaskortilla tai sovelluksen kautta liittymällä eri kategorioihin. Toistaiseksi myös sovelluksen yli voidaan jättää tarjouspyyntöjä, esim. sähköpostilla tai suullisesti.

Kategoria/Suosikit:

Asiakkaat voivat liittyä erilaisiin kategorioihin. Kategoriat ovat kiinnostuksen kohteita, mm. atk, taide, harrastukset, urheilu, tähtitiede jne.

Sähköinen mainos / e-mainos / e-tarjous:

HMTL -pohjainen sähköinen mainos, mikä voidaan lähettää asiakkaalle kanta-asiakassovelluksen sivulle.

Virtuaalinen kanta-asiakaskorttitunnus:

Virtuaalinen kanta-asiakaskortti koostuu puhelinnumerosta + tunnuksesta. Kanta-asiakaskortin voi tulostaa myös viivakoodina. Liikkeessä asiakas voi jättää yhteydenottopyynnön antamalla kanta-asiakastiedot mm. kanta-asiakaskortilla (puhelinnumero viivakoodina) ja syöttämällä näppäimistöllä tunnusluvun (virtuaalisen osan, mikä määrittää profiilin). Tiedot voidaan antaa myös suullisesti.

Tekniset lyhenteet:

XML (*extensible Markup Language*) on merkintäkieli tai standardi, jolla tiedon merkitys on kuvattavissa tiedon sekaan. XML-kieltä käytetään sekä formaattina tiedonvälitykseen järjestelmien välillä että formaattina dokumenttien tallentamiseen. XML-kieli on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomassoja selkeämmin.

Web-Service = **web service** on W3C:n määritelmän mukaan ohjelmistojärjestelmä, joka mahdollistaa keskenään yhteensopivan tietokoneiden välisen vuorovaikutuksen tietoverkon yli.

HTTP (Hypertext Protocol) = Yhteykäytäntö, jota käytetään WWW:ssä asiakkaan ja palvelimen välisiin yhteyksiin.

Palvelin (Server) = Tietokone, joka on määritelty vastaamaan verkosta tuleviin kutsuihin. Palvelinohjelma, joka vastaa HTTP-pyyntöihin.

Vastaus (Response) = Palvelimen hakupyynnön lähettämä ilmoitus asiakkaalle

Request = Hakupyyntö

URL = Uniform Resource Locator

1.4 Viitteet ja muut liittyvät dokumentit

viitatuut lähteet ovat tämän dokumentin lähdeluettelossa.

Muu määrittelydokumenttiin liittyvä tutkimus on Laurea-ammattikorkeakoulussa tehty opinnäytetyö. Tutkimusraportti sisältää kanta-asiakassovelluksen tutkimusongelman, lähtökohdat ja tavoitteet, menetelmien valinnat, työn käytännön tekemisen ja tutkimuksen yhteenvedon.

1.5 Yleiskatsaus dokumenttiin

Yleiskatsauksen tarkoituksena on esittää määrittelydokumentin rakennetta ja sisältöä. Dokumentissa on yhteensä kahdeksan kappalekokonaisuutta. Määrittelydokumentin sisältö on seuraavanlainen:

- ✓ Luvussa 2. kerrotaan kanta-asiakassovelluksen yleiskuvaa ympäristöstä, toiminnasta, käyttäjistä, oletuksista ja rajoitteista.
- ✓ Luvussa 3. kuvataan tiedon ja tietokannan rakennetta käsitelmän ja relaatiomallin avulla.
- ✓ Luvussa 4. kuvataan kanta-asiakassovelluksen toimintoja. Toiminnot on kuvattu käyttötapauskaavioiden ja tekstiselityksien avulla.

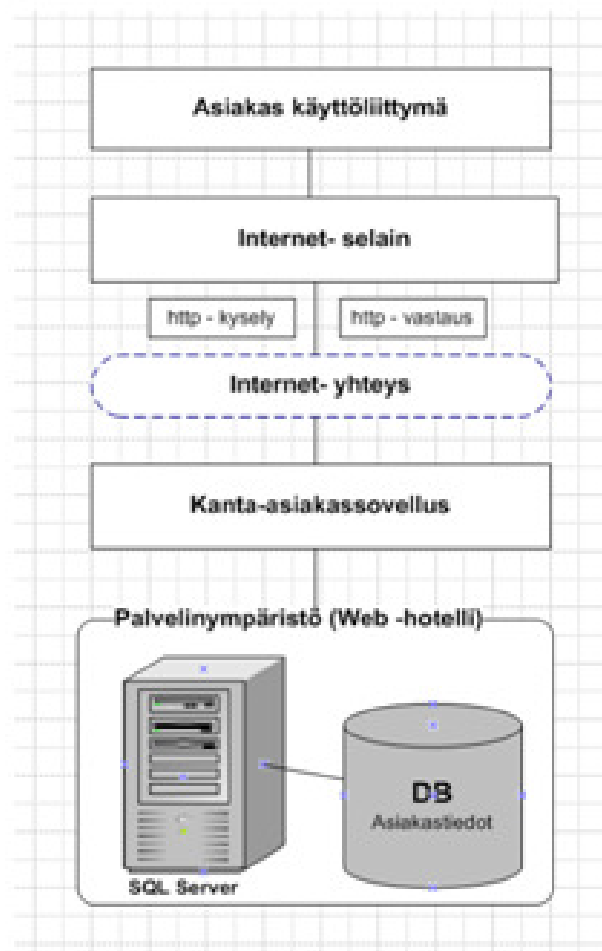
- ✓ Luvussa 5. kerrotaan kanta-asiakassovelluksen ulkoisista liittymistä, mm. sovelluksenkäyttöliittymän rakenteesta ja ulkoasusta sekä ohjelmisto-, laite- ja tietoliikenneliittymistä.
- ✓ Luvussa 6. kerrotaan muista ominaisuuksista, mm. tietoturva, toipuminen, ylläpito, suorituskyky ja operointi.
- ✓ Luvussa 7. kerrotaan suunnittelurajoitteista, mm. standardit, ohjelmisto- ja laiterajoitteista sekä muista rajoitteista.

2 Yleiskuvaus

Tässä luvussa on tarkoituksena antaa yleiskuva sovelluksen toiminnasta. Luvussa kerrotaan ympäristöstä, toiminnasta, käyttäjistä, yleisistä rajoitteista, oletuksista ja riippuvuuksista. Käyttäjä rooleja on hahmotettu käyttötapauskaaviolla ja käyttötapaus tekstiselityksin.

2.1 Ympäristö

Kanta-asiakassovelluksen toimintaympäristö on kuvattu alla olevassa kuvassa. Palvelinympäristönä on Web- hotelli, missä toimii Microsoftin SQL- tietokanta. Tiedonsiirto tapahtuu http- protokollan avulla. Selaimet välittävät kyselyjä (request) ja vastauksia (response) myyjän sekä asiakkaan välillä. Myyjät voivat ohjelmallisesti kysyä Web- service- palvelusta tai http- request -kyselyn avulla tietoja. Kehitysympäristönä on Windows-alusta ja ASP.NET ja ohjelma toteutetaan C# - kielellä.



Kuva 1: Sovelluksen toimintaympäristö

2.2 Toiminta

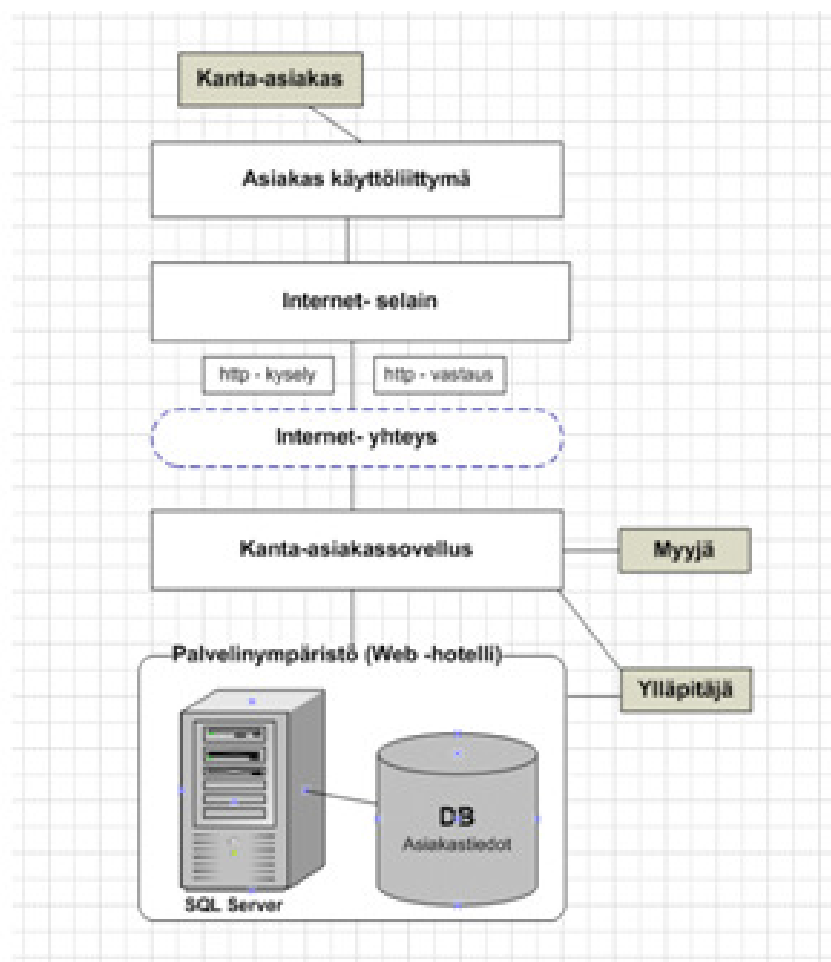
Kanta-asiakassovellusta käytetään Web -selaimen kautta. Selaimen kirjoitetaan kanta-asiakassovelluksen URL -kohdeosoite, jonka jälkeen avautuu Web-käyttöliittymä, mistä pääsee kirjautumaan sisään. Koska sovellus on Web - pohjainen, niin sen käyttäminen onnistuu mistä päin maailmaa tahansa ja milloin vain.

Tietokannan tarkoituksena on varastoida ja ylläpitää asiakastietoja. Tietoja on pystyttävä lisäämään, poistamaan, päivittämään, hakemaan ja muokkaamaan. Sovelluksen käyttäjien on pystyttävä kirjautumaan järjestelmästä sisään ja ulos. Ensimmäisellä käyttökerralla asiakkaan on rekisteröityvä kanta-asiakassovelluksen käyttäjäksi. Rekisteröitymisen jälkeen asiakas pääsee vapaasti käyttämään sovellusta ja muokkaamaan tietojaan.

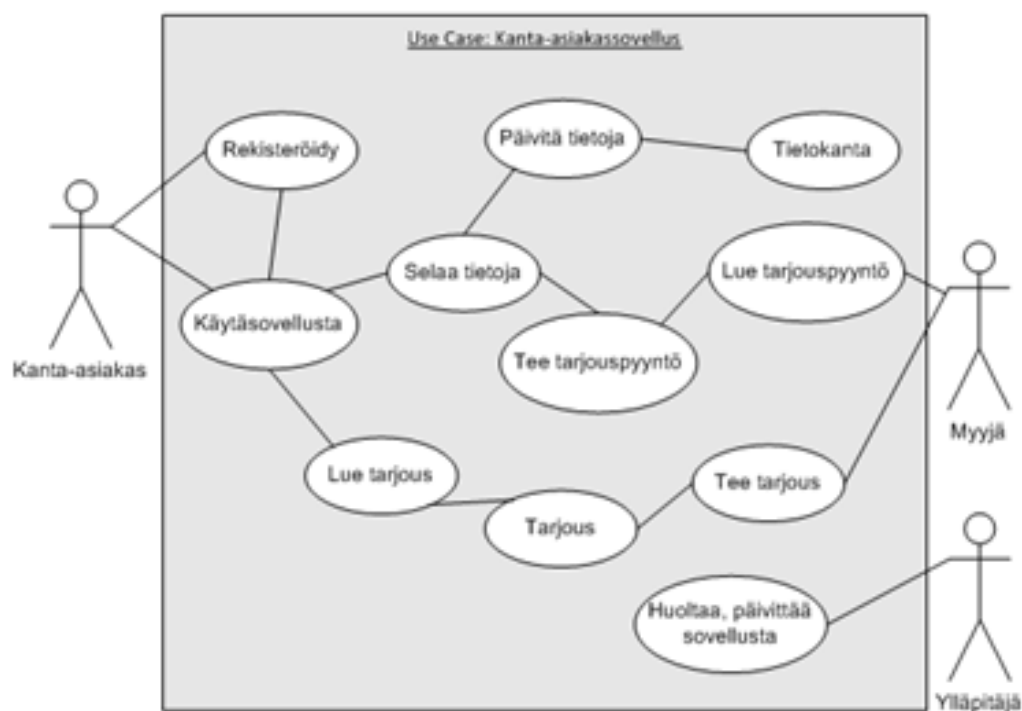
Kanta-asiakassovelluksesta on pystyttävä lähettämään tarjouspyyntöjä ja ottamaan vastaan tarjouksia. Kanta-asiakassovelluksen lokin tehtävänä on kirjata sovelluksessa tapahtuvat poikkeavuudet ja auttaa ongelmatilanteiden selvittämisessä.

2.3 Käyttäjät

Kanta-asiakassovelluksen käyttäjät ovat myyjä, asiakas ja ylläpitäjä (ks. kuvassa 2). Myyjillä ei ole oikeutta kirjautua sovellukseen. Myyjät voivat lähettää asiakkaille tarjouksia tuotteista tai palveluista tai vastaanottaa tarjouspyyntöjä. Asiakas on yksityinenhenkilö, joka ostaa myyjiltä tuotteita, ottaa vastaan tarjouksia/mainoksia ja tekee tarjouspyyntöjä. Ylläpitäjä auttaa järjestelmään liittyvissä ongelmatilanteissa ja ylläpitää kanta-asiakassovellusta.



Kuva 2: Sovelluksen toimintaympäristö + käyttäjät



Kuva 3: Käyttötapauskaavio käyttäjistä

Käyttötapaus 1: Tarjouspyynnön tekeminen myyjälle ja tarjouksen lähettäminen asiakkaalle

Käyttäjät:	Asiakas, Myyjä
Esiehdot:	Asiakkaan tietojen on löydyttävä järjestelmästä.
Toiminta:	<ol style="list-style-type: none"> Asiakas jättää myyjälle uuden tarjouspyynnön automaation kautta tai liikkeessä virtuaalisella kanta-asiakaskortilla. Myyjä ottaa vastaan tarjouspyynnön ja näkee asiakkaan sallimat profilitiedot. Profilitiedoissa on sallittu markkinointitapa ja

	<p>kiinnostuksen kohteet, minkä perusteella myyjä toimii.</p> <p>4. Myyjä kohdistaa asiakkaalle kiinnostuksen mukaan tarjouksen ja lähettää sen asiakkaan määrittelemällä markkinointitavalla.</p> <p>5. Asiakas vastaanottaa tarjouksen myyjältä.</p> <p>6. Asiakas tekee ostopäätöksen.</p>
Poikkeukset:	<p>P1. Yhteys tietokantaan on poikki.</p> <p>P2. Tarjouspyyntö on puutteellinen.</p>
Lopputulos:	Tarjouspyyntö ja tarjous on vastaanotettu onnistuneesti.

Taulukko 1: Käyttötapaus 1: tarjouspyyntö ja tarjous

Käyttötapaus 2: Tietojen päivittäminen: muokkaus, poisto, valinta, lisäys	
Käyttäjät:	Asiakas
Esiehdot:	Asiakastietojen pitää olla järjestelmässä. Asiakkaan pitää olla kirjautuneena omalla hallintasivulla.
Kuvaus:	<p>1. Asiakas selaa omia tietoja.</p> <p>2. Asiakas haluaa päivittää tiettyjä asiakas- tai profilitietoja ja tekee sen jälkeen tallennuksen.</p> <p>3. Muutokset menevät suoraan tietokantaan ja järjestelmä ilmoittaa onnistuneesta tietojen päivittämisestä.</p>
Poikkeukset:	<p>P1. Tietojen päivittäminen ei onnistunut. Pakolliset kentät pitää olla täytettynä ja tieto oikeassa muodossa.</p> <p>P2. Yhteys tietokantaan on poikki.</p>
Lopputulos:	Tietojen muutokset onnistuivat ja tallentuivat tietokantaan onnistuneesti.

Formlek Oy

Toiminnallinen määrittelydokumentti
Versio 1.0

15

Taulukko 2: Käyttötapaus 2: Tietojen päivittäminen

Käyttötapaus 3: Rekisteröityminen kanta-asiakkaaksi	
Toimijat:	Asiakas
Esiehtot:	Asiakkaan tiedot eivät ole järjestelmässä.
Kuvaus:	<ol style="list-style-type: none"> Järjestelmä näyttää rekisteröitymislomakkeen. Asiakkaan on täytettävä pakolliset ja tarvittavat tiedot lomakkeelle (Puhelinnumero, nimi, osoite, sähköposti, salasana). Järjestelmä tarkistaa rekisteröitymisen ja tallentaa uuden asiakkaan tiedot.
Poikkeukset:	<p>P1. Tiedot ovat virheelliset tai puutteelliset lomakkeella.</p> <p>P2. Asiakkaan puhelinnumerolla löytyy jo olemassa oleva asiakkuus.</p>
Lopputulos:	Asiakkaan tiedot on tallennettu kanta-asiakasjärjestelmään. Näin kanta-asiakkuus on luotu ja vahvistustiedot tulevat sähköpostiin.

Taulukko 3: Käyttötapaus 3: Rekisteröityminen



Käyttötapaus 4: Kirjautuminen sovellukseen	
Käyttäjät:	Asiakas
Esiehtot:	Asiakastietojen pitää olla järjestelmässä.
Kuvaus:	<ol style="list-style-type: none"> Järjestelmä avaa kirjautumissivun. Sovellukseen kirjaututaan antamalla käyttäjätunnus (puhelinnumero) ja salasana. Järjestelmä hakee asiakkaantiedot ja vertaa vastaavako

	<p>asiakkaan antamat tiedot tosiaan.</p> <p>4. Asiakas pääsee tarkastelemaan omia tietojaan.</p> <p>5. Asiakas poistuu lopuksi sovelluksesta kirjautumalla ulos</p>
Poikkeukset:	<p>P1. Käyttäjätunnus on oikein, salasana on väärin.</p> <p>P2. Käyttäjätunnus on väärin, salasana on oikein.</p> <p>P3. Käyttäjätunnus on väärin, salasana on väärin.</p> <p>P4. Asiakkaan tietoja ei löydy järjestelmästä.</p> <p>P5. Yhteys tietokantaan on poikki.</p>
Lopputulokset:	<p>Asiakas on kirjautunut järjestelmään ja pääsee hallitsemaan omia tietojaan. Asiakas pääsee kirjautumaan ulos</p>

Taulukko 4: Käyttötapaus 4: Kirjautuminen

2.4 Yleiset rajoitteet

- Yleistä markkinointiin liittyvää lainsäädäntöä noudatettava
- Asiakastiedot ovat luottamuksellisia (asiakasrekisteri)
- Tietoturvan huomioiminen kanta-asiakassovelluksen kehittämisessä
- Tietokantaa on pystyttävä käyttämään ajasta riippumatta
- Virhetilanteista ja poikkeuksista ilmoitukset
- Loki
- Tietokantaa on pystyttävä käyttämään usea käyttäjä samaan aikaan
- Ohjeet (ylläpitäjä/käyttäjä)

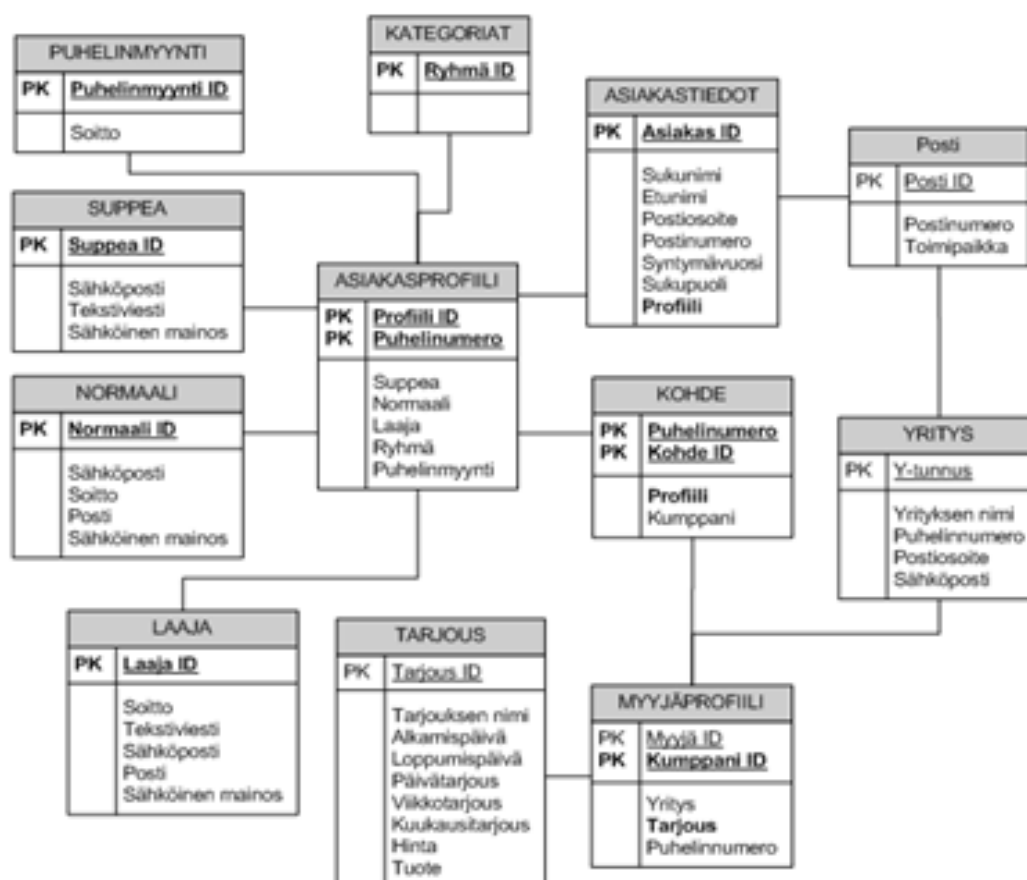
2.5 Oletukset ja riippuvuudet

- Windows- käyttöjärjestelmä
- Mahdollisimman tehokas palvelin
- Standardit
- Selaimet

3 Tiedot ja tietokanta

Kappaleessa kuvataan kanta-asiakassovelluksen tiedot ja tietokanta. Kappaleessa 3.1 esitetään tietokannan tiedot käsitelmällin avulla. Tietokannan rakenne ja ominaisuudet esitetään tarkemmin kappaleessa 3.2. Tietojen ominaisuudet on määritelty alustavasti. Lopulliset suunnittelut tehdään suunnittelu- ja toteutusvaiheessa.

3.1 Tietokannan rakenne ja taulujen väliset yhteydet



Formlek Oy

Toiminnallinen määrittelydokumentti
Versio 1.0

18

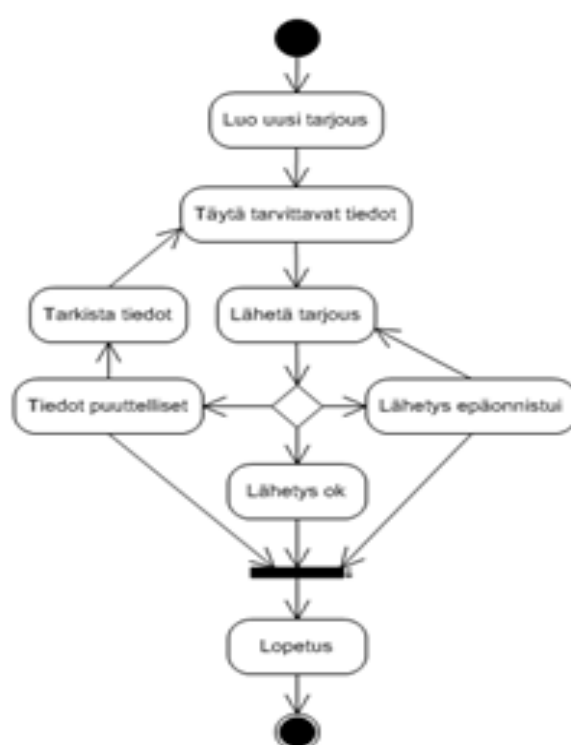
3.2 Tietokannan tarkempi kuvaus

Taulun nimi	Kentän nimi	Tietotyyppi	Pituus	Avain	Kuvaus
KATEGORIAT	Ryhmä ID	Luku	Autom.	Pääavain	Yksilö taulun tiedot.
	Suosikki	Teksti	20		Tieto suosikeista, mm. ATK, hanastukset, taide jne.
ASIAKASPROFILI	Profiili ID	Luku	Autom.	Pääavain	Yksilö profilitaulun tiedot
	Puhelinnumero	Numero	20	Pääavain	Yksilö profilitaulun puhelinnumeron, kahta samaa ei voi olla.
	Suppea	Teksti		Viteavain	Valinta: Kyllä tai ei
	Normaali	Teksti		Viteavain	Valinta: Kyllä tai ei
	Laaia	Teksti		Viteavain	Valinta: Kyllä tai ei
	Puhelinmyynti	Teksti		Viteavain	Valinta: Kyllä tai ei
	Ryhmä	Luku	5	Viteavain	Viittaa Kategoria taulun ryhmä ID:hen.
PUHELINMYynti	Puhelinmyynti ID	Luku	Autom.	Pääavain	Yksilö tiedot
	Soitto	Teksti			Valinta: Kyllä tai ei
SUPPEA	Suppea ID	Luku	Autom.	Pääavain	Yksilö taulun tiedot
	Sähköposti	Teksti			Valinta: Kyllä tai ei
	Tekstiviesti	Teksti			Valinta: Kyllä tai ei
	Sähköinen mainos	Teksti			Valinta: Kyllä tai ei
NORMAALI	Normaali ID	Luku	Autom.	Pääavain	Yksilö taulun tiedot
	Posti	Teksti			Valinta: Kyllä tai ei
	Soitto	Teksti			Valinta: Kyllä tai ei
	Tekstiviesti	Teksti			Valinta: Kyllä tai ei
	Sähköposti	Teksti			Valinta: Kyllä tai ei
LAAJA	Laaia ID	Luku	Autom.	Pääavain	Yksilö taulun tiedot
	Soitto	Teksti			Valinta: Kyllä tai ei
	Tekstiviesti	Teksti			Valinta: Kyllä tai ei
	Posti	Teksti			Valinta: Kyllä tai ei
	Sähköposti	Teksti			Valinta: Kyllä tai ei
	Sähköinen mainos	Teksti			Valinta: Kyllä tai ei
ASIAKASTIEDOT	Asiakas ID	Luku	Autom.	Pääavain	Jatkava luku, tiedon pituutta ei ole vielä määritetty.
	Sukunimi	Teksti	40		Asiakkaan sukunimi
	Etunimi	Teksti	40		Asiakkaan etunimi
	Postiosoite	Teksti	30		Asiakkaan osoite tieto.
	Postinumero	Luku	5		Postinumero, muotoa 00000, aina 5-numeroa.
	Sähköposti	Tekstiluku	40		Sisältää erikoismerkin: "@", tieto muotoa: xxxx@hotmail.xxx
	Syntymävuosi	Vuosisuku	4		4-numeroinen luku. Asiakkaan syntymävuosi.
	Sukupuoli	Teksti	5		Valitaan: mies tai nainen
	Profiili	Luku	Autom.		Jatkava luku, tiedon pituutta ei ole vielä määritetty.
POSTI	Posti ID	Luku	Autom.	Pääavain	Jatkava luku, tiedon pituutta ei ole vielä määritetty.
	Postinumero	Luku	5		Postinumero, muotoa 00000, aina 5-numeroa.
	Postitoimipaikka	Teksti	20		Postinumeron mukaan tallennettava paikkakuntatieto. Huom. Samalla postinumerolla voi olla useampi paikkakunta. ID viittaa

KOHDE	Kohde ID Puhelinnumero Profiili Kumppani	Luku Luku	Autom. 99	Paaevain Viiteevain	Yksilö jokaisen kohde riviin taulusta Profiilitaulusta viittaava puhelinnumero Asiakkaan profiili taulun tunniste, minkä myyjä saa ja sen mukaan näkee asiakkaan profiilin. Myyjän kumppani ID tiedon kulussa.
TARJOUS	Tarjous ID Tarjouksen nimi Aikamispäivä Loppumispäivä Viikkotarjous Päivätarjous Kuukausitarjous Hinta Tuote	Luku Teksti Päivämäärä Päivämäärä Aikaväli Päivämäärä Aikaväli Lukuluku Nimi	Autom. 40 10 20	Paaevain	Yksilö tarjoustaulun rivit. Tiedon pituutta ei määritely. Tarjoukseen liittyvä nimi, otsake Milloin tarjous alkaa Milloin tarjous päättyy Viikon pituinen tarjous, mikä ilmoitetaan aikavälillä. Päivän tarjoustuote. Yhden päivän voimassa oleva tarjous K:n voimassa oleva tarjous. Ilmoitetaan tarjouksen alk ja lop. Tarjoustuotteen hinta. Mikä tuote/tuotteet tarjouksessa
MYyjÄPROFIILI	Myyjä ID Kumppani ID Yritys Tarjous Puhelinnumero	Luku Luku Teksti Luku Luku	Autom. Autom. 30 Autom. 10	Paaevain Paaevain Viiteevain	Yksilö myyjän Kohde taulun kautta siirtyvä tunniste. 4-numeroinen luku. Yrityksen nimi, missä myyjä työskentelee Viittaava tarjoustaulusta. Myyjän puhelinnumero.
YRITYS	Y tunnus Yrityksen nimi Puhelinnumero Postiosoite Sähköposti Postinumero	Luku Teksti Luku Teksti Tekstiluku Luku	9 30 10 30 40 5	Paaevain	Sisältää erikoismerkin "~", tieto on muodossa: 0000000-0 Kertoo yrityksen nimen. Yrityksen yhteystieto puhelinnumero, max. 10-numeroa. Yrityksen osoite/sjainti tieto Sisältää erikoismerkin "@", tieto muotossa: xxx@yritys.xx Postinumero, muotoa 00000, aina 5-numeroa

4 Kanta-asiakassovelluksen toiminnot

Tässä luvussa on kuvattu sovelluksen toimintoja. Toimintojen kuvaukseen on käytetty apuna tilakaavioita. Kanta-asiakassovelluksen perustoiminnot ovat: kirjautuminen sisään ja ulossovelluksesta, tietojen päivittäminen (muokkaus, poistaminen ja lisäys) rekisteröityminen, tarjouksien ja tarjouspyyntöjen lähettäminen.



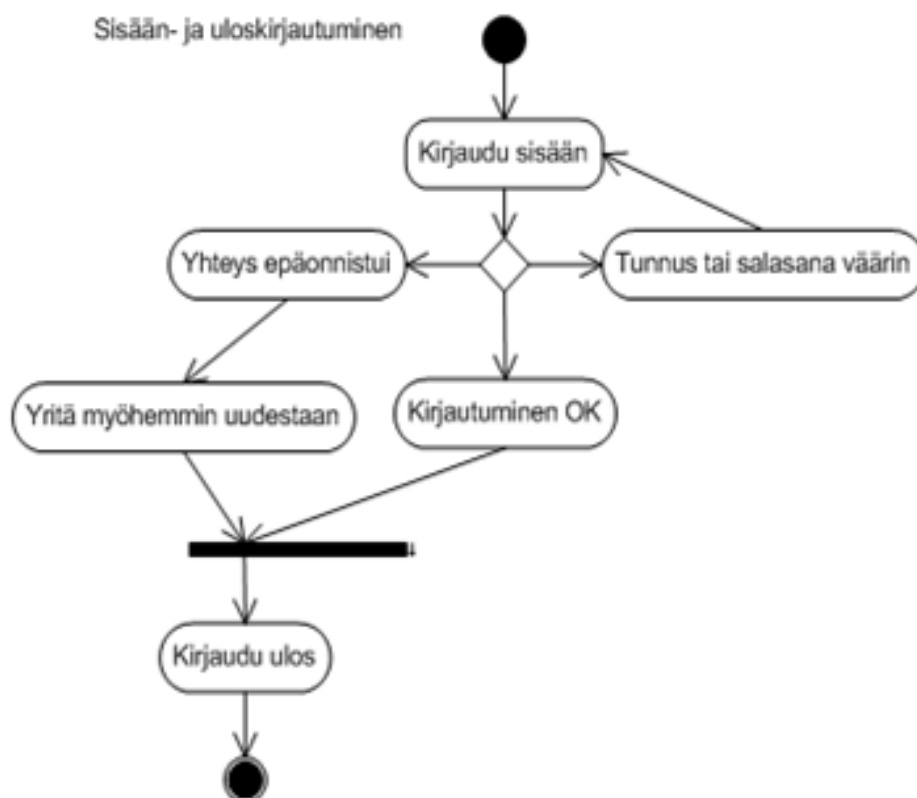
Kuva 4: Tarjouksen lähettäminen

4.1 Sisään ja uloskirjautuminen

Kirjautuminen onnistuu, jos asiakas on tehnyt rekisteröitymisen. Kirjoitetaan käyttäjätunnus sekä salasana. Sen jälkeen painetaan "kirjautu sisään" -painiketta. Kirjautuminen voi epäonnistua, jos tietokantaan ei saada yhteyttä. Kirjautuminen ei onnistu, jos jompikumpi tai kummatkin käyttäjätunnus/salasana on väärin. Kirjautuminen ei onnistu, jos käyttäjätunnus tai salasana puuttuu tekstiriviltä. Kuvasta 5 puuttuu myös mahdollisuus pyytää **unohtunut salasana**. (HUOM.! tämä kohta otettava myös huomioon suunnittelussa ja toteutuksessa).

Toiminto:

Sisään- ja uloskirjautuminen

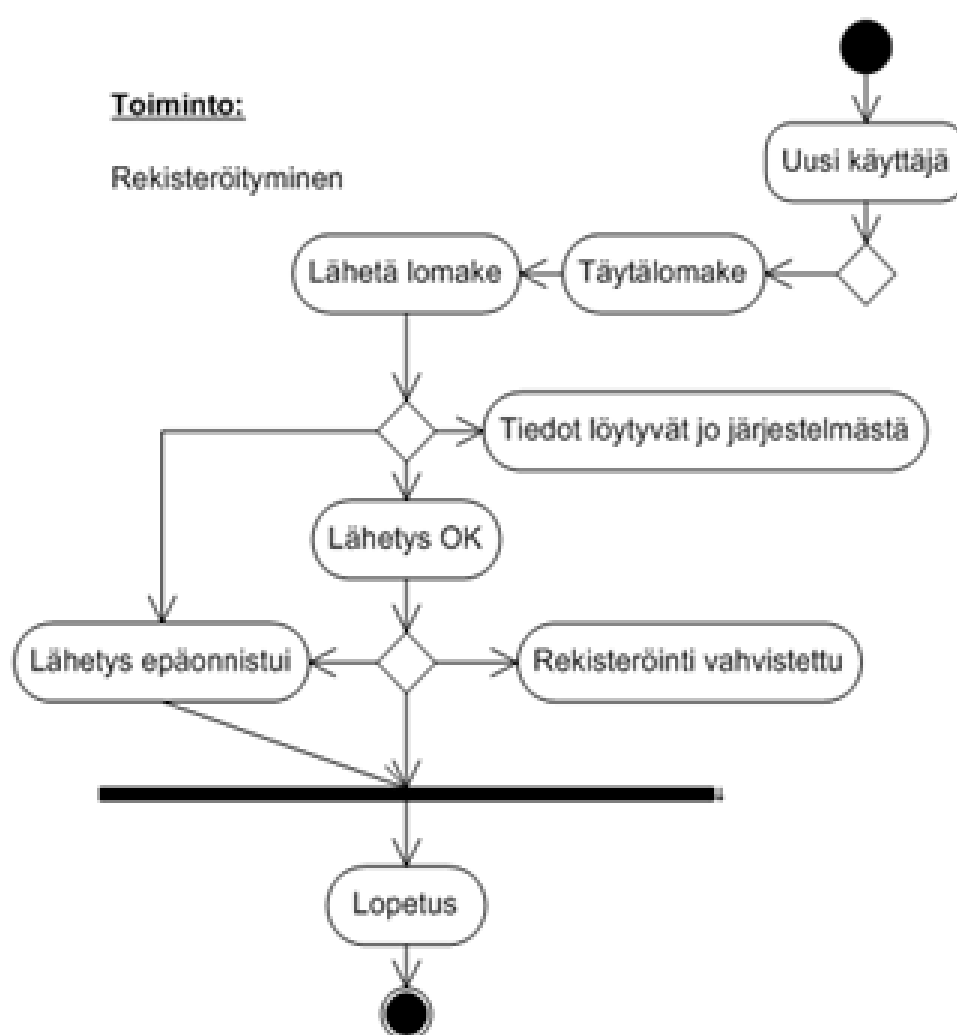


Kuva 5: Sisään ja uloskirjautuminen

4.2 Rekisteröityminen

Sovelluksen käyttäminen vaatii rekisteröitymisen. Rekisteröityneen käyttäjän on oltava uusi käyttäjä. Uusi käyttäjä täyttää rekisteröintilomakkeen ja lähettää sen. Rekisteröinti voi epäonnistua, jos yhteystietokantaan on epäkunnossa. Rekisteröinti ei onnistu, jos tiedot löytyvät jo järjestelmästä. Lomakkeen lähettäminen ei onnistu, jos lomakkeella on puutteellisia tietoja (tämä kohta puuttuu kuvasta 6). Tietojen

onnistuneesta lähettämisestä tulee ilmoitus ja vahvistus rekisteröitymisestä (sähköpostiin/puhelinnumeroon). Kuvauksesta puuttuu selite, siitä että käyttäjä voi pyytää **uuohtunutta salasanaa**, ettei sen takia tarvitse tehdä uutta rekisteröintiä (HUOM.! tämä otettava myös huomioon suunnittelussa ja toteutuksessa).



Kuva 6: Kanta-asiakassovelluksen toiminto rekisteröityminen

4.3 Tietojen päivittäminen

Kirjautumisen jälkeen sovelluksessa voidaan selata, lisätä, päivittää tietoja. Tietojen päivittämisen (muokkaus, poistaminen, lisääminen) jälkeen tulee aina ilmoitus onnistuneesta tallennuksesta. Tietojen päivittämisen epäonnistumisesta tulee ilmoitus myös.

Toiminto:

Tietojen päivittäminen (poisto, muokkaus, lisäys)



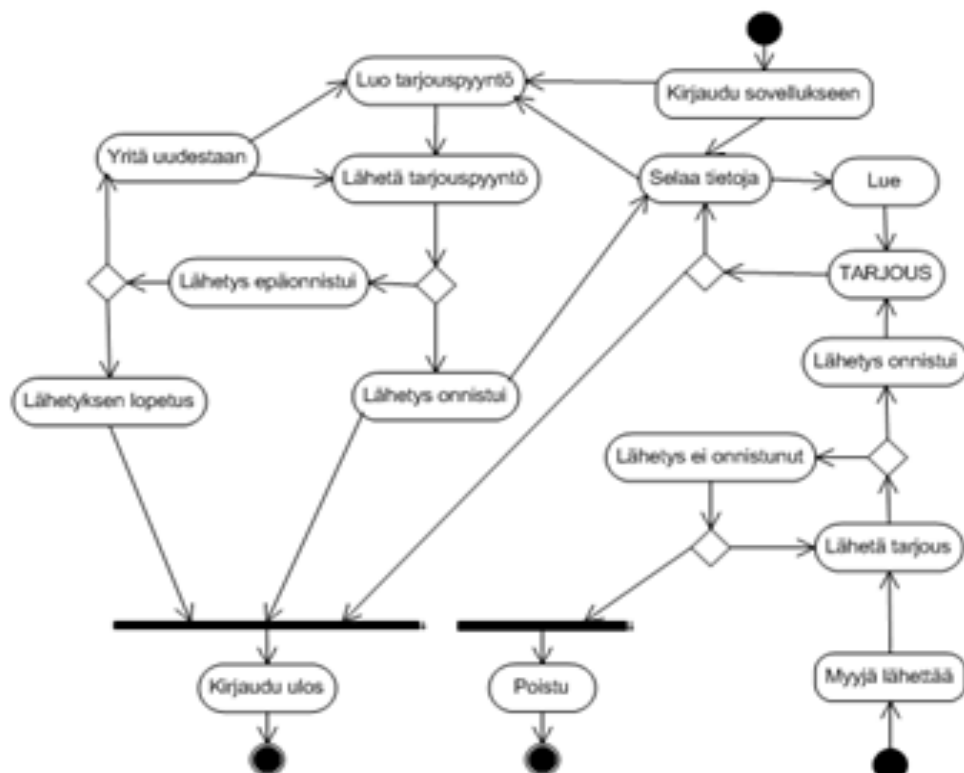
Kuva 7: Kanta-asiakassovelluksen toiminto tietojen päivittäminen

4.4 Tarjouspyynnön ja tarjouksen lähettäminen

Vain kanta-asiakas (rekisteröitynyt käyttäjä) voi lähettää tarjouspyynnön sovelluksen kautta. Tarjouspyyntö luodaan lomakkeella, johon täytetään pakolliset tiedot. Tarjouspyyntö lähetetään, jonka jälkeen saadaan kuittaus onnistuneesta lähettämisestä. Jos lähetys on tehty onnistuneesti tulee siitä ilmoitus. Tilanteen mukaan voi yrittää uudelleen lähettämistä tai keskeyttää lähettämisen. Tarjouspyyntölomaketta ei pysty lähettämään, jos siitä puuttuu pakollisia arvoja. Myyjä lähettää asiakkaalle tarjouksen asiakastietojen mukaan. Lähetys voi epäonnistua, jos tietokantaan ei saada yhteyttä tai sen sisältö ei syystä x siirry onnistuneesti. Onnistuneen lähetyksen jälkeen asiakas pääsee lukemaan tarjouksen sisällön. Sen jälkeen voi jatkaa tietojen selaamista tai poistua järjestelmästä.

Toiminto:

Tarjouksien ja tarjouspyyntöjen lähettäminen



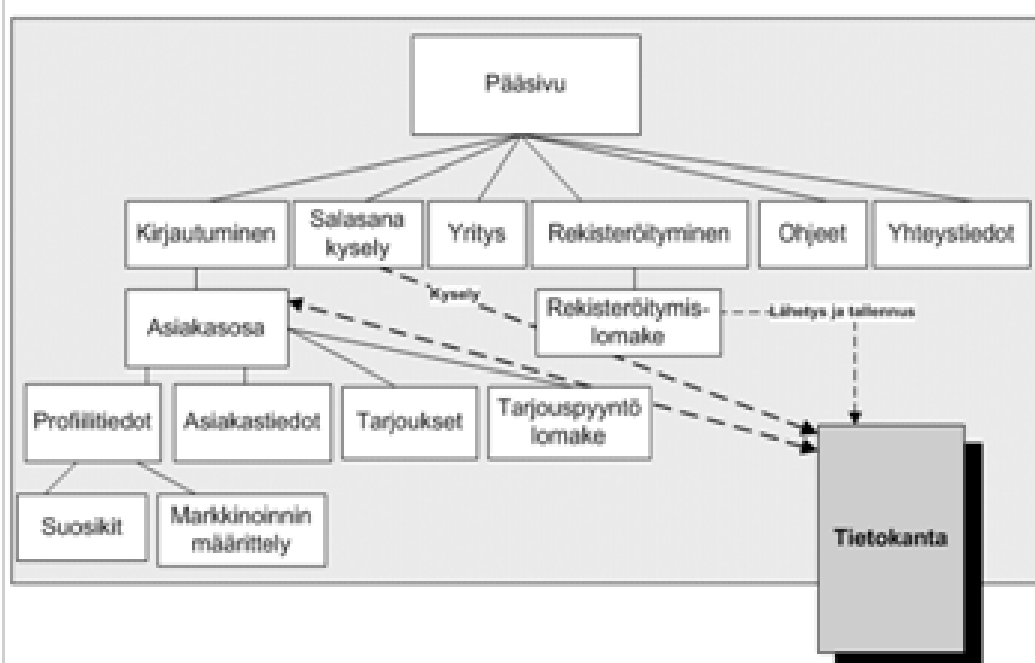
Kuva 8: Kanta-asiakassovelluksen toiminto tarjouksien hallinta**5 Ei-toiminannalliset vaatimukset****5.1 Käyttöliittymä**

on oltava helposti opittavissa, virheetön, käyttäjäystävällinen ja helpokäyttöinen. Käyttöliittymää on pystyttävä käyttämään nopeasti ja sen on oltava ulkonäöltään selkeä. Käyttöliittymän tulee olla helppo käyttää, ettei käyttäjän tarvitse miettiä, mitä mistäkin kohdasta tapahtuu. Käyttöliittymän suunnittelussa on käytettävä yleisesti tunnettuja ja hyväksi havaittuja menetelmiä. Käyttöliittymän suunnittelussa on otettava huomioon käyttäjien tarpeet ja tavat. Käyttöliittymässä pitää olla ohjesivusto tai ohjaavia "velhoja, sanoja".

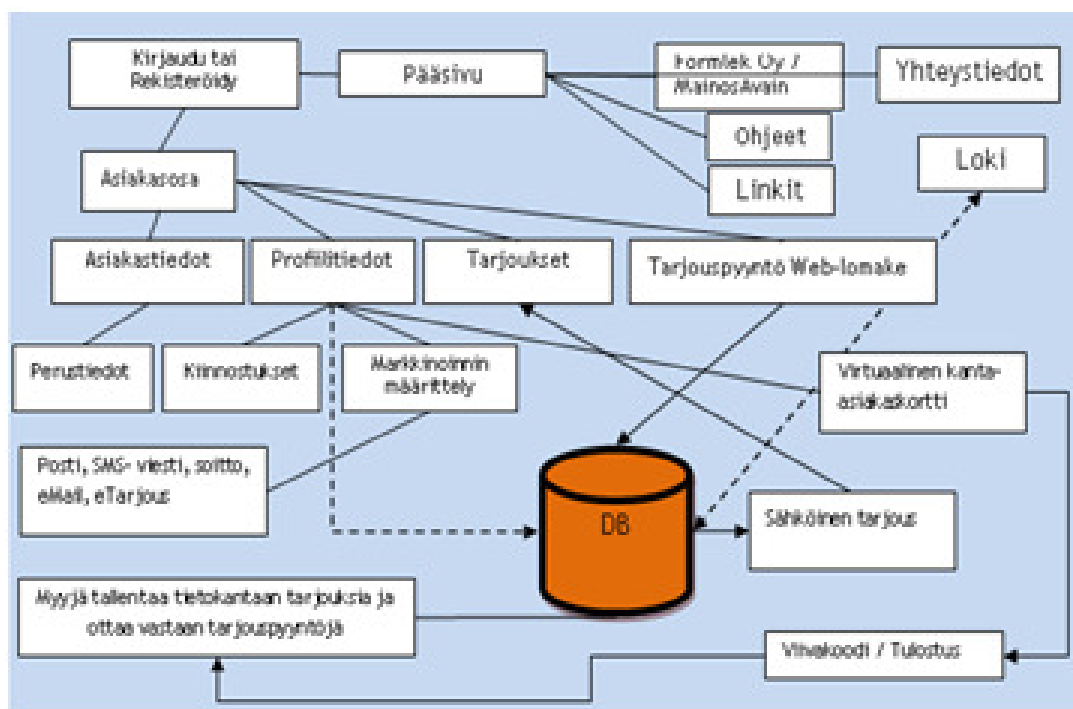
Vaatimuksia käytettävyyteen ja käyttöliittymän suunnitteluun:

- ✓ Helposti opittavissa ja muistettavissa
- ✓ Käyttöliittymän suunnittelussa ja toteuttamisessa sovelletaan yleisesti tunnettuja ja hyväksi havaittuja suunnittelu päätöksiä
- ✓ Älä pakota käyttäjää tekemään mitään ylimääräistä
- ✓ Sivustosta kevyt
- ✓ Ei turhia intro - sivuja
- ✓ Sisällön oltava asiallista ja käyttäjää kiinnostavaa
- ✓ Lyhyttä ja ytimekästä tekstiä
- ✓ Silmäilyn helpottavia otsikoita
- ✓ Älä julkaise sivuja keskeneräisenä; suunnittele julkaisu järjestys
- ✓ Tee yhtenäinen navigointi
- ✓ Erotta navigointi selkeästi muun sivuston rakenteesta
- ✓ Sivujen latausaika pidettävä mahdollisimman alhaisena

5.1.1 Kanta-asiakassovelluksen sivustorakenne



Kuva 9: Sivusto rakenne 1



Kuva 10: Sivusto rakenne 2

5.1.2 Alustavat käyttöliittymän komponentit

Valtuutettu Markkinoija

Sulje 










Valtuutetun markkinoijan avulla voit hallita markkinointitapoja ja tehdä tarjouspyyntöjä tuotteista sekä vastaanottaa tarjouksia.

Kirjaudu sovellukseen:

Minulla ei ole tunnusta? [Rekistroidy täällä.](#)

Salasana unohtunut? [Vaihda salasanasi täällä.](#)

Kuva 11: Protomalli kirjautumis -sivusta

1	Nimi	Osa	Toiminto	Tapahtuma	Toiminto OK	Toiminto Virhe	Kuva
2	Lähetä	Painike	Painallus	Lähetää tiedon.	Tietojen lähettäminen onnistui.	Tiedon lähettäminen epäonnistui.	
3	Tallenna	Painike	Painallus	Tallentaa.	Tietojen tallennus onnistui.	Tallentaminen epäonnistui.	
4	Kirjautu ulos	Painike	Painallus	Poistuu käyttöliittymästä.	Istunto suljettu.	Poistuminen ei onnistunut.	
5	Kirjautu sisään	Tekstirit+ Painike	Syöte+ painallus	Ensinnä kirjoitetaan Käyttötunnus ja salasana, sen jälkeen painetaan Kirjautu sisään.	Kirjautuminen ok, kun pääsee sisään sovellukseen etusivulle, missä on terehdys.	Tunnus tai salasana väärin. Tunnusta tai salasanaa ei löydy järjestelmästä. Tyhjä tieto. Yhteys epäonnistui. Salasana unohtunut ?	Ks. Kuva
6	Poista	Painike	Painallus	Poistaa tietoja.	Tietojen poistaminen tehtiin onnistuneesti.	Tietojen poistaminen ei onnistunut. Valitse	
7	Peruta	Painike	Painallus	Peruttaa toiminnon.	(Toiminnon nimi) keskeytetty / perutettu.	Perutus ei onnistunut.	
8	Muokkaa	Painike	Painallus	Muokkaa tietoja.	Tietojen muokkaaminen onnistui.	Tietojen muokkaus ei onnistunut. Tarkista tiedot.	
9	OK	Painike	Painallus	Kuitaus painike.	Painetaan OK - vaimennus.	Toiminto epäonnistui.	
10	Rekisteröityminen	Lomake	Syöte	Täytetään lomakkeen pakolliset tiedot ja lähetetään.	Lähetys onnistui. Rekisteröinti vahvistettu.	Lomakkeesta on puuttellisia tietoja. Tarkista tiedot. Tiedot löytyvät jo järjestelmästä. Lähetys epäonnistui.	
11	Tajouspyyntö	Lomake	Syöte	Täytetään ja lähetetään.	Kaikki pakolliset tiedot täytetty.	Lomakkeesta on puuttellisia tietoja. Tarkista tiedot.	
12	Suosikit	Valintaruutu	Valinta	Valitaan suosikki kategorioita.	Valitaan haluttu kohde ja sen jälkeen painetaan tallenna.	Valitse suosikki kategoria ennen tallennusta. ET ole valinnut yhtään suosikkia.	
13	Virheellinen toiminto	Ikon	Ilmoitus	Ilmoittaa käyttäjän tekemästä virheestä.	Ilmoittaa onnistuneesti virheilmoituksen.	Sivulla tapahtui virhe, ole hyvä ja tee X. (riippuu virheilänteestä).	

Kuva 12: Käyttöliittymän alustavia komponentteja

Formlek Oy	Toiminnallinen määrittelydokumentti	30
	Versio 1.0	

Täytä rekisteröintilomake ja lähetä lomake:
Pakolliset tiedot on merkitty (*) :llä

(*)

(*)

(*)

(*)

(*)

(*)

Valitse itsellesi sopiva käyttätunnus ja salasana:

(*)

(*)

(*)

Kuva 13: Protomalli rekisteröitymislomakkeesta

5.2 Laitteistoliittymät

Kanta-asiakassovelluksen laitteistoliittymiä ovat:

- ✓ kannettava ja pöytäkone
- ✓ näppäimistö ja hiiri
- ✓ viivakooditulostin
- ✓ pieni syöttönäppäimistö (numeronäppäimistö)
- ✓ kortinlukija
- ✓ monitori

Viikooditulostimen on tarkoituksena tulostaa asiakkaan puhelinnumero viivakoodin muotoon. Asiakas voi tulostaa itselleen viivakooditulosteen, minkä voi antaa liikkeessä tai automaatilla. Näppäimistö on sitä varten, että asiakas pääsee antamaan ensin viivakoodilukijalle/kortinlukijalle viivakoodinsa, minkä jälkeen asiakas syöttää tunnusluvun. Tämän perusteella myyjä saa asiakkaan profiilitiedot. Tietokannasta määräytyy asiakkaan valitsema profiili (suppea, normaali, laaja, puhelinmyynti) mihin on määritelty valtuutettu markkinointikeinot.

5.3 Ohjelmistoliittymät

Kanta-asiakassovelluksen on oltava aina käytettävissä (7/24h). Sitä varten tullaan tekemään erillinen ylläpitosuunnitelma, jossa huomioidaan järjestelmän ylläpitokatkokset sekä järjestelmän nopeasiirtäminen varakoneelle.

Virhetilanteen sattuessa kanta-asiakassovelluksesta on saatava selkeä virheilmoitus syntyneestä ongelmatilanteesta. Mikäli kyse on poikkeavasta toiminnosta pyritään siitä kertomaan käyttäjälle, mitä pitäisi tehdä, esim. "osoitetiedot ovat puutteelliset, ole hyvä ja tarkista osoitetiedot". Mikäli kantasovelluksessa tapahtuu virhetilanne on siitäkin saatava ilmoitus asiakkaalle, esim. "tietokantaan ei saada tällä hetkellä yhteyttä, ole hyvä ja kokeile myöhemmin uudelleen".

Lokitiedoston tarkoituksena on pitää listaa sovelluksessa sattuneista virhetilanteista. Lokitiedossa on käytävä ilmi, milloin virhe on tapahtunut, virheilmoituksen selite ja tietysti olisi hyvä pitää jotakin ajan tasalla olevaa virhelistaa, mistä voi löytää ratkaisun ongelmaan nopeasti.

5.4 Turvallisuus ja suojaukset

Tietoturvan ja suojauksen toimivuus tulee olemaan haastavaa kanta-asiakassovelluksen kehittämisessä. Toimivassa suojausmenettelyssä suojaustasoja on useampi, niiden yhteistoimivuudella saadaan sovellukselle kokonaissuoja. On

huolehdittava, että tietoturvasot ovat ajan tasalla ja hoitavat niille tarkoitettuja tehtäviä. Tietokannan varmistukset ja tietokannat hoidetaan Web- hotellin kautta. Huomioonotettavia tietoturva asioita kanta-asiakassovelluksen kehittämisessä:

- ✓ Palomuurin ohjattava käyttäjät oikeaan palveluun
- ✓ Käyttöjärjestelmällä on oltava palvelurajoituksia
- ✓ WWW - palvelulla on oltava tunnistautumismenettelyt ja turva-asetukset kohdillaan
- ✓ Kanta-asiakassovelluksen alustalla on suojausmenetelmät
- ✓ Tietokantaan on käyttörajoitukset ja aineisto on suojattu
- ✓ Kanta-asiakassovelluksen käyttäjän tunnistautuminen pakollinen ja riittävän vahva (käyttöoikeuden tarkistaminen)
- ✓ Tunnistautumisen määräytyminen sovelluskohtaisesti (selaimen kautta kirjautumispyyntö palveluun, eväste (passport authentication)???)
- ✓ Mihinään sovellustiedostoihin ei saa päästä käsin WWW - palvelun kautta
- ✓ Myyjät pääsevät tietoihin vain kutsujen kautta

5.5 Siirrettävyys ja yhteensopivuus

Kanta-asiakasjärjestelmä tullaan toteuttamaan C#- ohjelmointikielellä ja Web-palvelin on ASP.NET. Sovelluksen alustana on Windows- käyttöympäristö, joten kanta-asiakassovelluksen on oltava siirrettävissä vastaavanlaiseen käyttöympäristöön. Tutkittava miten kanta-asiakassovellus toimii eri selaimilla. Lähinnä olisi hyvä kartoittaa ne selain versiot, mitä vähintään pitää olla käyttäjällä käytössä, jotta sovellus toimii ilman ongelmia. Toimivuutta on seurattava selain päivitysten yhteydessä.

5.6 Ylläpidettävyys

Laadun varmistamiseksi käyttöliittymä tehdään moduuleista ja luokista, jolloin toiminnan parantaminen ja muutokset ovat helposti hallittavissa. Ohjelmassa on hyvä

kommentointi ja selkeät kuvaavat muuttujien ja funktioiden nimet. (Tavoitteena, että jokaisen funktion lähdekoodi voidaan tulostaa A4-kokoiselle paperille)

5.7 Operointi (Käyttäjän ylläpitotoimet)

Käyttäjiä voisi ohjeistaa:

- ✓ WWW - selaimien asetuksista ja ominaisuuksista, jotta saavat sovelluksen tehokkaasti ja turvallisesti käyttöön (välimuisti, evästeet, JavaScript jne.)
- ✓ Palomuurin ja virustorjunnan suosittele
- ✓ Käyttöohje sivustolla /sovelluksessa
- ✓ Mahdollisimman uusi käyttöjärjestelmä (Windows)

6 Suunnittelu rajoitteet

6.1 Standardit

Standardi:**Selite:**

CR	C Sharp = Ohjelmointikieli
AJAX	Asynchronous JavaScript And XML
ASP.NET	Active Server Pages
SQL	Structured Query Language
ANSI/IEEE830-1998	Toiminnallinen määrittely

6.2 Laitteistorajoitteet

Mahdollisimman tehokas palvelin (ei ole vielä määritelty tallennustilan vientiä ja vapaata työtilan määrää). Ei ole määritelty asiakkaan kannalta laitteistorajoitteita (vähintään Windows NT - Windows 7 - järjestelmiä pyörittävät koneet). Toimiva

modeemi/mobiililaajakaista tietoliikenneyhteyttä varten: kaapeli, ADSL tai datapaketti.

6.3 Ohjelmistorajoitteet

- ✓ Microsoft kehitysympäristö, Visual Studio 2008 tai 2010

- ✓ Tietokanta: MySQL
- ✓ Palvelimen käyttöjärjestelmä: Windows 7
- ✓ Web - hotelli. Palvelun tarjoaja vielä avoimena (Nebula)
<http://www.nebula.fi/webhotels.php>

HUOM.! →

Testausvaiheessa on testataan sovelluksen toimivuutta erilaisissa käyttöympäristöissä (asiakkaan kannalta). Testaan kanta-asiakassovelluksen toimivuus

käyttöjärjestelmillä: Windows (NT, 2000, XP, Vista, Windows 7), Linux (jakelut x) ja Mac Os (versiot x). Jokaisessa on oltava vähintään selaimet Mozilla tai Explorer (myös toimivuutta voidaan testata Safarilla ja Netscape:llä, jos aikaa riittää testaukseen).

On suositeltavaa käyttää mahdollisimman uutta selainta:

- ✓ Microsoft Internet Explorer 7.0 tai parempi
- ✓ Netscape Navigator
- ✓ Firefox Mozilla
- ✓ Google Chrome
- ✓ Apple Safari
- ✓ Opera

Lähteet

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. 8. painos. Pieksämäki: RT-Print.

WWW -dokumentti. Luettu 2010.

<<http://www.cs.tut.fi/~projekti/dokumentit/ylla-sisalto.txt>>

Kuvat

Kuva 1: Sovelluksen toimintaympäristöntaympäristö.....	10
Kuva 2: Sovelluksen toimintaympäristö + käyttäjät.....	12
Kuva 3: Käyttötapauskaavio käyttäjistä.....	13
Kuva 4: Tarjouksen lähettäminen	20
Kuva 5: Sisään- ja uloskirjautuminen.....	21
Kuva 6: Kanta-asiakassovelluksen toiminto rekisteröityminen	22
Kuva 7: Kanta-asiakassovelluksen toiminto tietojen päivittäminen	23
Kuva 8: Kanta-asiakassovelluksen toiminto tarjouksien hallinta	25
Kuva 9: Sivuston rakenne 1	26
Kuva 10: Sivuston rakenne 2.....	27
Kuva 11: Protomalli kirjautumis -sivusta.....	28
Kuva 12: Käyttöliittymän alustavia komponenttejä.....	29
Kuva 13: Protomalli rekisteröitymislomakkeesta.....	30

Taulukot

Taulukko 1: Käyttötapaus 1: tarjouspyyntö ja tarjous.....	14
Taulukko 2: Käyttötapaus 2: Tietojen päivittäminen.....	15
Taulukko 3: Käyttötapaus 3: Rekisteröityminen.....	15
Taulukko 4: Käyttötapaus 4: Kirjautuminen	16