

Maansiirtoyrityksen uuden tietojärjestelmän konseptointi



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinna, Tietojenkäsittelyn koulutusohjelma

Kevät, 2017

Elias Koskinen

Tietojenkäsittelyn koulutusohjelma
Hämeenlinna

Tekijä	Elias Koskinen	Vuosi 2017
Työn nimi	Maansiirtoyrityksen uuden tietojärjestelmän konseptointi	
Työn ohjaaja/t	Tommi Saksa	

TIIVISTELMÄ

Työn toimeksiantajana oli Kuljetus Vilkinen Oy Rengosta. Yrityksen lähinnä kuormakirjoihin liittyvää paperityötä haluttiin vähentää uuden sovelluksen avulla. Työn tarkoituksena oli selvittää, mitä ominaisuuksia maansiirtoyrityksen tietojärjestelmä tarvitsee ja mitä voidaan jättää pois. Sovellus koostuu kahdesta osasta, kuljettajien sovelluksesta mobiililaitteella ja toimiston työpöytäsovelluksesta.

Teoriaosassa käydään läpi käytettävyyttä käsitteenä. Sen jälkeen tutustutaan erilaisiin tekniikkoihin, joista suurinta osaa käytetään varsinaisessa suunnittelutyössä. Rautalankamallit, iterointi ja vaatimusmäärittely käsitellään ensin. Sitten tutustutaan käyttötapauksiin ja käyttäjätarinoihin, joista tarinat päättyivät käytännön osuuteen.

Suunnittelutekniikoina käytettiin vaatimusmäärittelyä, käyttäjätarinoita ja rautalankamalleja. Ensin tehtiin vaatimusmäärittely, josta saatiin työlle tavoitteet. Seuraavaksi kirjoitettiin muutamia käyttäjätarinoita selkeyttämään sovelluksen käyttötarkoitusta. Viimeisenä tulivat varsinaiset konseptikuvat. Asiakkaan kanssa piirrettiin suunnitelmakuvia paperille, jotka myöhemmin muutettiin rautalankamalleiksi. Rautalankamalleja iteroitiin muutaman kierroksen verran. Lopputuloksena oli kirjallisia ja kuvallisia konsepteja sovelluksesta.

Työssä selvitettiin, mitä ominaisuuksia maansiirtoyritys tarvitsee sovellukseltaan ja mitä ominaisuuksia voidaan jättää pois yksinkertaisuuden, hinnan ja helppokäyttöisyyden nimissä. Työn tulokset voivat olla yleistettävissä myös muihin sovelluksiin.

Avainsanat rautalankamalli, vaatimusmäärittely, iterointi, konsepti

Sivut 29 sivua

Degree Programme in Business information technology
Hämeenlinna

Author	Elias Koskinen	Year 2017
Subject	Concepting information system for earthmoving company	
Supervisors	Tommi Saksa	

ABSTRACT

The client for this thesis was an earthmoving company Kuljetus Vilkinen Oy from Renko, Finland. Client wanted a new software to create and manage consignment notes, which had developed to a time-consuming task. The goal of the project was to create concept pictures and find out what features were necessary to include in the application. Application consists of two parts, mobile application for drivers and desktop version for office.

The theory part handles usability as a wider concept. It is followed by introduction to different planning techniques, which are later used in the practical part. Wireframe model, Iteration and requirements analysis are handled first, followed by use cases and user stories, from which user stories were selected for final work.

Requirements analysis was implemented first to find out the goals of the software. User stories were written next to clarify the usage and purpose of the software. Lastly came actual concept drawings. They were first designed on paper with client and later turned into wireframe models. Models were then iterated for a few cycles to develop them.

One of the goals was to determine which software features are necessary for earthmoving company and which can be dropped out for the sake of simplicity and price. The results of this thesis can be generalized to include other software.

Keywords Wireframe model, iteration, requirements analysis, concept

Pages 29 pages

SISÄLLYS

1	JOHDANTO.....	1
2	KÄYTTÖLIITTYMÄSUUNNITTELU.....	2
2.1	Käytettävyys.....	2
2.2	Rautalankamalli.....	3
2.3	Vaatimusmäärittely.....	4
2.3.1	Toiminnalliset ja ei-toiminnalliset vaatimukset.....	5
2.3.2	Vaatimusmäärittelyprosessi.....	5
2.4	Käyttötapaukset (use case).....	6
2.5	Käyttäjätarinat (User story).....	7
2.6	Iterointi.....	8
2.7	Balsamiq.....	9
3	TESTAUS.....	10
4	TOTEUTUSSUUNNITELMA.....	12
5	VAATIMUSMÄÄRITTELY.....	14
5.1	Tavoitteiden määrittely.....	14
5.2	Tarpeiden tunnistus.....	14
5.3	Jalostetut vaatimukset.....	15
6	KÄYTTÄJÄTARINAT.....	16
6.1	Kuljettajat.....	16
6.2	Toimisto.....	16
7	VARHAISET KONSEPTIT.....	18
7.1	Lähtötilanne – kuormakirja.....	18
7.2	Luonnokset.....	18
7.3	Ensimmäiset rautalankamallit.....	19
8	RAUTALANKAMALLIT.....	21
8.1	Mobiilisovellus.....	21
8.2	Vaihtoehtoinen mobiilisovelluskonsepti.....	25
8.3	Toimistosovellus.....	27
9	YHTEENVETO.....	30
	LÄHTEET.....	31

1 JOHDANTO

Opinnäytetyön toimeksiantajana on Kuljetus Vilkkinen Oy Rengosta. Yrityksellä on alle kymmenen työntekijää ja paljon kalustoa, kuten kuorma-autoja, kaivinkoneita ja tiehöyliä. Yrityksen toimenkuvaan kuuluu erilaisien maa-ainesten, soran, lumen ja asfaltin kuljetusten lisäksi mm. maanrakennusta ja tietöitä. Sovelluksen tärkein toiminto on kuormakirjojen tuotto ja arkistointi.

Yritys haluaa toimintaansa helpottamaan sovelluksen, joka toimii sekä mobiililaitteilla kuljettajien mukana, että toimistolla työpöytäsovelluksena. Työpöytäsovelluksella voisi seurata mobiililaitteilla tuotettua dataa ja hallita kuormakirjoja. Tässä opinnäytetyössä tarkoituksena on suunnitella sovelluksen toimintaa ja luoda konseptipiirroksia.

Tarkoituksena on ideoida selkeä ja helppokäyttöinen sovellus juuri maansiirtoyrittäjien erityistarpeisiin. Sovellukseen halutaan vain oleelliset ominaisuudet, jotta siitä saadaan mahdollisimman yksinkertainen ja edullinen. Työn tarkoitus onkin selvittää, miten yksinkertaiseen voidaan mennä, jotta mobiilisovellus pystyy saamaan tarvittavat tiedot kuljettajalta.

Työn alkuosassa käydään läpi käyttöliittymä- ja sovellussuunnittelun perusteita ja menetelmiä, joita sitten käytetään sovelluksen suunnitteluun ja konseptikuvien luontiin. Tekniikoista käydään läpi rautalankamallit, iterointi, vaatimusmäärittely, testaus, käyttötapaukset ja käyttäjätarinat.

Käytännön osiossa käytetään teoriaosiossa läpikäytyjä tekniikoita sovelluksen suunnitteluun. Osio sisältää paljon rautalankamalleja selityksineen. Kuvat ovat työn varsinaista tuotetta, joiden avulla asiakas saa paremman kuvan siitä mitä haluaa. Työn avulla voi myös helpommin selittää mitä on tilaamassa kolmannelta osapuolelta. Asiakas voi saamiensa mallikuvien avulla lähettää tarjouspyyntöjä sovelluskehittäjille.

2 KÄYTTÖLIITTYMÄSUUNNITTELU

Käyttöliittymällä tarkoitetaan niitä osia sovelluksesta tai laitteesta, jotka käyttäjä näkee ja joihin hän pääsee vaikuttamaan itsenäisesti (Immonen 2002). Käyttöliittymä on suunniteltu käyttäjää varten ja se yritetään tehdä mahdollisimman luontevaksi. Käyttöliittymä koostuu esimerkiksi valikoista, napeista, säätimistä ja tekstikentistä.

Myös käyttöliittymän visuaalisuus on tärkeää, varsinkin mobiilisovelluksissa. Erilaiset symbolit auttavat hahmottamista varsinkin älypuhelimissa, joissa tila on rajallinen ja jokaiseen painikkeeseen ei kannata kirjoittaa mitä ne tekevät. Symbolien merkitys on myös helppo muistaa sovellusta uudelleen käytettäessä. Monet symbolit ovat universaaleja eli ne tarkoittavat lähes samoja asioita eri sovelluksissa. Esimerkiksi hammasratas symboloi asetuksia ja suurennuslasi hakutoimintoa. Mobiilikäyttöjärjestelmiin, kuten Androidiin, tuotetuille sovelluksille on olemassa paljon suosituksia, joilla pyritään yhdenmukaistamaan sovellusten ulkoasua ja toimintoja tietyllä alustalla, jotta käyttäjän olisi helpompi oppia uusien sovellusten käyttö (Android n.d).

2.1 Käytettävyys

Sinkkonen, Kuoppala, Parkkinen ja Vastamäki (2002, 19) kertovat tutkijoiden ja ISO-standardoinnin mukaan käytettävyyden määrittävän ainakin seuraavien tekijöiden mukaan: opittavuus, virheettömyys, muistettavuus, tehokkuus, miellyttävyys ja tuottavuus. Kirjoittajat toteavat määritelmien olevan kuitenkin vajavaisia ja käytettävyyden sisältävän paljon muutakin.

Kuten monesti tieteellisesti asioita määriteltäessä, muuttuvat yksinkertaiset asiat monimutkaisiksi. Käytettävyys terminä on selkeä ja useimmilla tulee jo pelkästä sanasta mielikuva siitä mitä se voisi olla. Moni osaa varmaankin sanoa jonkun sovelluksen, verkkosivun tai muun tuotteen, joka on joskus ärsyttänyt hankalalla käytettävyydellään. Huono käytettävyys syö käyttäjän halua suorittaa tehtäviä tuotteen avulla. Pahimmillaan tämä tarkoittaa sitä, että käyttäjä lopettaa tuotteen käytön ja siirtyy kilpailijan asiakkaaksi. Käyttäjillä on kuitenkin usein melko hyvä sietokyky ja korkea kynnys vaihtaa sovellusta. Pieniä ärsyttävyyksiä siedetään, sillä vaihtaminen ja uuden opettelukin on hankalaa (Krug 2006, 18-19). Työelämässä huono käytettävyys laskee työtehoja ja tuottavuutta, kun taas vaikkapa sairaalamaailmassa se voi olla jopa hengenvaarallista.

Asiantuntijoille tehtävissä sovelluksissa voidaan käyttää ammattisanastoa, mutta varmintä on kuitenkin aina käyttää selkokieltä. Suunnittelijalle selvät termit eivät sitä ehkä ole asiakkaalle tai tuotteen loppukäyttäjälle.

Suunnittelussa, kuten rautalankamallissa, käytetyt termit voivat epähuomiossa jäädä lopulliseen tuotteeseen vaikeuttaen sen käyttöä (Sinkkonen ym. 2002, 35).

Käytettävyyden voidaan katsoa olevan HCI:n (Engl. Human-Computer Interaction) eli ihmisen ja tietokoneen vuorovaikutusta tutkivan tieteen alalaji. HCI:ssä ihmistä ei kuitenkaan ajatella organisaation osana tai tahtovana tekijänä, vaan pikemminkin vain toimintoja suorittavana osana koneistoa. (Sinkkonen ym. 2002, 20)

2.2 Rautalankamalli

Rautalankamalli on yksinkertaistettu kuvaus ohjelman toimintalogiikasta ja toimintojen sijoittelusta. Se on mustavalkoinen kuva, jossa näyttönäkymä on jaettu kenttiin. Niihin on kirjoitettu lyhyt kuvaus kunkin kentän sisällöstä tai toiminnasta (Kuva 1). Kenttien sijoittelulla voidaan jo hahmotella myös tulevan ohjelman ulkoasua, ottamatta kantaa muuhun ulkoasuun, kuten värimaailmaan. Rautalankamallissa ei ole kuvia tai logoja, vaan niiden kohdalla voisi lukea ”kuva”, ”kuva yrityksen henkilöstöstä”, ”logo” tai ”Facebook-logo”. Dynaamisten objektien, kuten alasetovalikoiden, toiminta yritetään myös kuvata tässä vaiheessa (Sinkkonen ym. 2009, 203). Rautalankamallien etu on niiden yksinkertaisuus ja tuoton nopeus. Suunnittelija voi piirtää useita ehdotuksia, joista asiakkaan kanssa karsitaan huonoimmat pois.



Kuva 1. Esimerkki rautalankamallista. Kentissä on kerrottu lyhyesti niiden toiminta.

2.3 Vaatimusmäärittely

Vaatimusmäärittely on tietoteknisen hankinnan tai sovelluskehitysprosessin ensimmäisiä vaiheita. Se on kirjallinen selonteko siitä, mitä asiakas haluaa sovellukselta. Myös tuotteen hinta kuuluu vaatimusmäärittelyn piiriin. Vaatimusmäärittely on hyvin tärkeä osa sovelluskehitysprosessia, sillä sen avulla saadaan sekä asiakkaalle, että kehittäjälle sama kuva halutusta sovelluksesta. Usein asiakkaat eivät edes aivan varmasti tiedä mitä he haluavat ja tarvitsevat ammattilaista kysymään oikeat kysymykset. Hyvällä vaatimusmäärittelyllä vältetään myöhemmältä hämmennykseltä, joka aiheutuu yhteisymmärryksen puutteesta.

Vaatimusmäärittely keskittyy vain siihen mitä tulevalta järjestelmältä vaaditaan, riippumatta teknisistä yksityiskohdista. Teknillisten asioiden raja voi kuitenkin olla häilyvä. Esimerkiksi asiakas voi vaatia sovelluksensa pyörivän Android-käyttöjärjestelmällä, mutta harvemmin ottaa kantaa vaikkapa palvelinarkkitehtuuriin. Vaatimusmäärittely saa usein alkunsa vanhojen järjestelmien tutkimus- ja kehittämishankkeista (Tieke 2005).

2.3.1 Toiminnalliset ja ei-toiminnalliset vaatimukset

Vaatimukset voidaan jakaa toiminnallisiin ja ei-toiminnallisiin. Toiminnalliset vaatimukset kuvaavat selkeitä ja tarkkaan määriteltyjä tapahtumia. Esimerkkinä voisi olla seuraava: *Asiakas voi kirjautua sisään tai ylläpitäjä voi poistaa käyttäjätilejä*. Näistä tapahtumista voidaan myöhemmin tuottaa käyttötapauksia, joita käsitellään seuraavissa luvuissa.

Ei-toiminnalliset vaatimukset voidaan jakaa kahteen eri aliluokkaan (Helsingin yliopiston tietojenkäsittelytieteen laitos, ohjelmistotuotannon luento 3 n.d.), laatuvaatimuksiin ja toimintaympäristön rajoitteisiin. Laatuvaatimukset ovat järjestelmän kokonaisuutta ohjaavia ominaisuuksia, kuten käytettävyys, suorituskyky ja tietoturva.

Toimintaympäristön rajoitteet taas koskevat fyysistä ja sosiaalista käyttöympäristöä sekä laitteistoa (Sinkkonen, Nuutila, Törmä 2009, 49). Toimintaympäristön rajoitteina on yksinkertaisimmillaan se, että nettisivun tai sovelluksen on toimittava hyvin myös älypuhelimien pienellä kosketusnäytöllä. Yrityksen laitteisto asettaa myös rajoitteita palvelimien ja tietoliikenneyhteyksien kapasiteetin muodossa. Fyysisen ja sosiaalisen ympäristön rajoitteita voisivat esimerkiksi olla vaikkapa laitteen käyttöympäristön kylmyys tai käyttäjän sokeus.

2.3.2 Vaatimusmäärittelyprosessi

Valmiiseen vaatimusmäärittelyyn johtava prosessi voidaan jakaa kolmeen selkeästi erilaiseen alueeseen. Ensimmäinen vaihe on *tavoitteiden määrittely*, jossa selvitetään mitä uudelta järjestelmältä halutaan ja perustellaan syitä miksi se pitäisi hankkia tai vanhaa järjestelmää päivittää. Hyviä perusteluja voisivat olla vaikkapa kasvanut asiakasmäärä, kilpailussa mukana pysyminen tai työprosessin tehostaminen. (Tieke, 2005)

Toisena vaiheena Tieke mainitsee *tarpeiden tunnistuksen*. Tässä vaiheessa tavoitteena on tunnistaa ja kerätä ensimmäisen vaiheen pohjalta käyttäjäkunnan tarpeita kolmatta vaihetta varten. Käyttäjät eivät välttämättä osaa kertoa, mitä sovellukselta oikeasti odottavat. Tekniikka kehittyy ja tavallinen toimistotyöläinen ei osaa edes kaivata uusimpia teknisiä helpotuksia. Toisaalta käyttäjillä voi olla päällekkäisiä tai jopa ristiriitaisia tarpeita. Samaa tehtävää varten ei tarvita useita toimintoja. Joku taas voi haluta pää-

syn johonkin tiedostoihin, jotka toinen haluaa pitää mahdollisimman salaisina. Tässä vaiheessa myös selvitetään, voiko mahdollista olemassa olevaa järjestelmää laajentaa tai sen osia muuten hyödyntää vai pitääkö koko järjestelmä rakentaa alusta uudestaan. On myös tärkeää muistaa liittymäkohdat muihin järjestelmiin. Omia rajapintoja ei voi muuttaa liikaa, sillä niiden tulee edelleen toimia asiakkaiden ja yhteistyökumppaneiden järjestelmien kanssa. (Tieke, 2005)

Kolmantena vaiheena tulevat varsinaiset *vaatimukset*. Toisesta vaiheesta saatujen tarpeiden pohjalta karsitaan ja tarkennetaan lopulliset vaatimukset. Jalostetuista vaatimuksista koostetaan kattava kuva tulevasta järjestelmästä. (Tieke, 2005)

2.4 Käyttötapaukset (use case)

Käyttötapaukset ovat yksi UML-kaaviokokonaisuuden kaavioista. Käyttötapaukset on usein johdettu vaatimusmäärittelyn toiminnallisista tapahtumista. Käyttötapaus on todellinen tai kuviteltu tapahtuma, jossa profiloitu käyttäjä käyttää ohjelmaa jonkin toiminnon suorittamiseen. Käyttötapaukset eivät sisällä tarkkaa teknistä kieltä ja niiden tulisi olla maallikon ymmärrettävissä (Bridging the gap 2017). Käyttäjiä (engl. Actor) on erilaisia, eritasoisilla oikeuksilla ja kyvyillä varustettuna ja he näkevät sovelluksen eri tavalla. Tyypillisin käyttötapaus on sovelluksen ladannut uusi käyttäjä. Tavallisella käyttäjällä on erilaiset oikeudet ja tarpeet kuin vaikkapa järjestelmää ylläpitävällä henkilöllä. Myös tietokanta tai muu tietokone voi olla käyttötapauksen käyttäjänä.

Käyttötapauksista piirretään kaavio, jossa jokaisen käyttäjän ja järjestelmän välisiä vuorovaikutustapahtumia kirjataan miellekartan tapaan (Kuva 2). Eri käyttäjien toimintoja yhdistellään viivoilla, jotta saadaan selville heidän välisensä vuorovaikutussuhteet. Näistä vuorovaikutussuhteista syntyy rajapintoja sovelluksen käyttäjäkuntien välille.



Kuva 2. Yksityiskohta erään kaappasovelluksen käyttötapauskaaviosta. Itsetehty.

Käyttötapausten tulisi olla vain yksi tapahtuma. Esimerkiksi käyttötapaus *"Käyttäjä kirjautuu palveluun löytääkseen ystävänsä profiilin"* voidaan jakaa kahteen osaan: *"Käyttäjä kirjautuu palveluun"* ja *"Käyttäjä hakee profiileja"*. Käyttötapauksissa ei pitäisi esiintyä teknistä sanastoa, kuten *"tekstikenttä"* tai edes *"klikata"*. Tämän vuoksi käyttötapauksessa ilmenneen tapahtuman voi toteuttaa sovelluksessa monella eri tavalla.

Myöhempisiin vaiheisiin jää teknisten yksityiskohtien suunnittelu. Aina sovelluksessa ei ole, eikä edes tulisi olla, vain yhtä tapaa suorittaa haluttu toiminto. Näitä vaihtoehtoisia käyttötapoja voi ilmestyä myös tahattomasti käyttäjien käyttäessä sovellusta tavoilla, joita suunnittelija ei osannut edes ajatella (Krug 2006, 26). Esimerkkinä Krug käyttää tapauksia, joissa hän on nähnyt käyttäjien kirjoittavan täydellisiä URL-osoitteita hakukoneiden tekstikenttiin. Tämä ei varsinaisesti ole väärin, mutta tehotonta. Käyttäjä on kuitenkin täysin tyytyväinen sovelluksen toimintaan.

2.5 Käyttäjätarinat (User story)

Käyttäjätarinoita kutsutaan joskus myös skenaarioksi (Stellman 2009). Niissä kuvataan laajemmin muutaman virkkeen pituisesti käyttäjän tarpeita. Yhdestä käyttäjästä on useita tarinoita eri tilanteisiin. Stellman neu-

voo, että käyttäjätarinat tulisi aina kirjoittaa sellaisella kielellä ja termistöllä, jotta kuka tahansa käyttäjä ymmärtää välittömästi mistä tarinassa on kyse.

Esimerkkitarina: Käyttäjä avaa ohjelman ja huomaa saaneensa viestin. Hän siirtyy viesteihin ja valitsee saapuneen viestin. Käyttäjä lukee viestin ja vastaa siihen. Käyttäjä lisää liitetiedostoksi kuvan ja tekstidokumentin.

Käyttäjätarina ei siis ota kantaa teknisiin yksityiskohtiin vaan käyttäjän tarpeisiin, siihen mitä sovelluksella pitäisi pystyä tekemään. Yksittäinen käyttäjätarina ei kuitenkaan kata kaikkia sovelluksen toimintoja, vaan vain ne mitä kussakin tarinassa tarvitaan. Tästä johtuen käyttäjätarina ei ole työkalu koko sovelluksen kokonaisuuden kuvaamiseen.

2.6 Iterointi

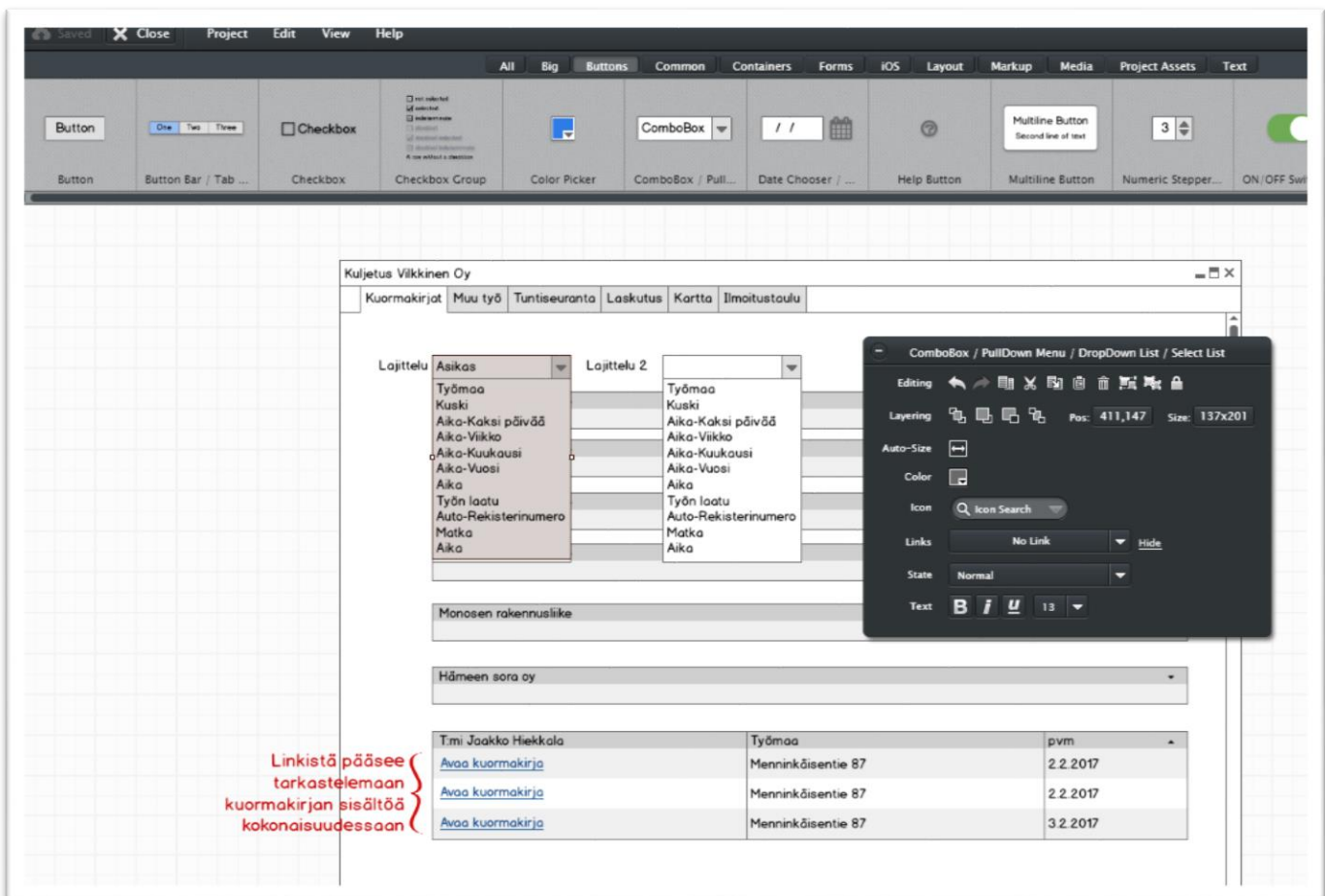
Kirjallisen määritelmän mukaan *iteroinnilla* tarkoitetaan syklistä tuotekehitystä, jossa toistetaan riittävän monta kertaa sama prosessi, jotta saadaan tyydyttävä konsepti tuotteesta. Vaiheet ovat tuotteen nykytilan arviointi, yksityiskohtien lisäys, prototyypin teko ja testaus. Yksinkertaisten prototyyppien syklit voivat olla niin nopeita, että saman testauspäivän aikana voidaan tehdä useita kierroksia saman testaajan kanssa samasta aiheesta. (Sinkkonen ym. 2009, 204-205.)

Tässä tapauksessa prototyyppi tarkoittaa jotain osaa koko sovelluksesta tai vaikkapa vain kuvaa. Kuvien teko onnistuu helposti tarkoitukseen suunnitelluilla ohjelmilla, kuten Balsamiqilla. Prototyypeillä hyvin samankaltainen rooli kuin rautalankamalleilla. Prototyyppien voidaan kuitenkin sanoa olevan seuraava askel, jossa on mukana toiminnallisuuksia. Prototyypeillä havainnollistetaan vaikeasti selitettäviä asioita tuotteen tilaajalle tai loppukäyttäjille tai vain varmistetaan, että puhutaan samoista asioista ja kaikki ymmärtävät. (Sinkkonen ym. 2009, 204-205.)

2.7 Balsamiq

Balsamiq on erilaisten sovellusten toiminnan mallinnukseen ja rautalankamallien tekemiseen tarkoitettu ohjelma (Kuva 3). Sillä voidaan tuottaa toimivia malleja, joissa napit vievät eri sivuille ja tekstikenttiin voi kirjoittaa. Balsamiq sisältää peruselementtejä niin mobiili- kuin työpöytäsovelluspuolelta.

Balsamiqin etuna on juuri mallien todenmukainen visuaalisuus. Suullisesti ja kuvien avulla selitettäessä monet kohdat voivat jäädä epäselviksi. Jos asiakas ei ymmärrä jotakin toimintoa, hän saattaa vain ajatella, että ”kyllä se varmaan oli ihan ok” ja ”kyllä suunnittelija tietää mitä tarvitaan”.



Kuva 3. Balsamiqin työnäkymä. Kuvaruutukaappaus.

3 TESTAUS

Testauksen perimmäinen idea on se, että omalle työlleen tulee sokeaksi. Vaikka olisi suunnitellut sovelluksia kauan, aina voi tulla virheitä ja oma ratkaisu johonkin toimintoon tai visuaalisuuteen voi olla huono. (Sinkkonen ym. 2009, 297.) Myös käyttäjäkunnan erityistarpeet vaativat usein testaamista, sillä sovelluksen suunnittelija ei voi olla kaikkien alojen ammattilainen. ”Näin on aina tehty” -ajattelumalli ei ole toiminut hengästyttävää vauhtia kehittyvän tietotekniikan ja siihen liittyvien alojen kanssa.

Sovellusta tulee testata kesken kehitysprosessin, jotta virheitä voidaan karsia ennen niiden paisumista. Jos järjestelmän olennaisia toimintoja perustuu virheelliseen tai ei-toivottuun ominaisuuksiin, tulee niiden päälle rakennettujen järjestelmien korjaamisesta hankalaa ja kallista. Yhden henkilön testaaminen alussa on parempi kuin suuren joukon testaaminen lopussa (Krug 2006, 134). Ohjelma tulisi testata sen tulevalla käyttäjäkunnalla, kuten yrityksen työntekijöillä ja toimitusjohtajalla.

Jos aluksi testataan suurella joukolla ohjelman toimivuutta, virheitä ei löydy paljoa eli tehosuhte laskee. Jos sata ihmistä löytää saman, ohjelman toimintaa olennaisesti haittaavan virheen, on siinä 99 ihmistä turhaan. Iso virhe voi estää testaajia löytämästä muita virheitä, jotka voivat olla ”piilossa” ison virheen takana tai muuten huomaamattomampia. Esimerkiksi jos nappi ohjaa väärään paikkaan, ei tarkoitetulla kohdenäkymällä olevia virheitä pääse näkemään, sillä käyttäjällä ei ole keinoa löytää niitä. Jos samaa sovellusta olisi testattu vaikkapa kahdella pienemmällä ryhmällä prosessin aikana, olisi ensimmäinen joukko löytänyt isot virheet. Ne olisi korjattu ja seuraava ryhmä voi löytää uudet ja pienemmät virheet (Krug 2006, 138-139). Prosessi muistuttaa soran seulomista, jossa pienin kiviaines (pienet virheet) valuvat alimmas siivilöitten läpi ja isommat kivet jäävät ylempiin siivilöihin silmäkoon mukaan. Tietotekniikassa kuitenkin jonkin virheen korjaaminen voi synnyttää toisen, jopa suuremman, virheen. Tämän takia tietokonesovelluksen testaaminen on erityisen tärkeää prosessin varrella.

Krugin (2006, 139) mukaan koehenkilöiden ominaisuuksilla ei ole testauksen kannalta suurta merkitystä. Hänen mukaansa ennen käytettiin aivan liikaa aikaa tarkasti mm. iän, sukupuolen, koulutuksen ja taitojen mukaan profiloitujen koehenkilöiden etsimiseen. Sinkkonen, Nuutila ja Törmä suosittelivat profilointia, mutta kuitenkin maltillisissa määrin (2009, 81).

Vaikka tuote olisi tarkoitettu alan ammattilaisille, tulisi sen olla niin helppoksi ja selkokieliseksi tehty, että maallikkokin voisi sitä käyttää. Ihmiset todennäköisemmin pahoittavat mielensä siitä, että tuote on epäselvä, kuin että heidän asiantuntijuuttaan epäillään liian helpolla kielellä. Usein riittää, että vaikkapa Android-sovellusta testaamaan valittu henkilö on joskus tätä

käyttöjärjestelmää pyörittävää laitetta käyttänyt vaikkapa kuvien ottamiseen, verkon selailuun ja pelaamiseen. Suurin osa käyttäjistä on kuitenkin keskivertoja, sillä he eivät pysy kauaa aloittelijoina ja kestää kauan tulla todella taitavaksi (Vibor Cipan 2010). Täytyy muistaa, että uuden sovelluksen käytössä kaikki ihmiset ovat yhtä lailla aloittelijoita, teknisestä osaamisesta huolimatta.

4 TOTEUTUSSUUNNITELMA

Tässä työssä tavoitellaan kattavaa kuvaa asiakkaan haluamasta sovelluksesta. Työssä tuotetaan paljon konseptikuvia sovelluksesta, ulkoasusta ja toiminnoista. Haasteena on toimeksiantajayrityksen monipuolisen toimenkuvan sovittaminen yhteen sovellukseen. Yritys tekee erilaisten maasora ja asfalttikuljetusten lisäksi satunnaisia lumitöitä ja kaivinkoneurakointia.

Asiakas tavoittelee sovelluksella tehostusta yrityksensä toimintaan varsinkin kuormakirjojen täytön ja hallinnan osalta. Asiakas kokee yrityksensä kasvun johdosta, että hänellä ei ole enää aikaa muuhun kuin toimistotöihin ja yrityksen asiakkaiden ja työntekijöiden asioiden selvittelyyn. Sovellus toisi helpotusta kuormakirjojen tuottamaan suureen paperityömäärään ja työntekijöiden käsialan tulkkaamiseen. Yhdenmukaisuutta saadaan kuormakirjojen täytön standardisoinnilla ja keskitetyllä hallinnalla ja arkistoinnilla. Kuormakirjojen ollessa sähköisessä muodossa, ne eivät myöskään katoa tai jää ohjaamon lattialle pyörimään.

Sovelluksen on tarkoitus myös auttaa työn kustannusten arvioinnissa ja hallinnassa. Sovellus kerää dataa esimerkiksi kuljetusmatkoista ja kuormien painoista tai kuutiomääristä. Kuljettajien sovelluksesta saadaan myös kunkin kuljettajan kullekin työmaalle tai asiakkaalle tekemästä tunti- ja kuormamäärästä. Näillä tiedoilla voidaan luoda tilastotietoja, joista on apua yrityksen toiminnan suunnittelussa.

Kuljettajat voivat ilmoittaa sovellukseen auton rekisterinumeron perusteella korjaustarpeita. Työpöytäsovellus hoitaa myös laskujen hallintaa. Asiakas voi tutkia, jäsenellä ja tulostaa laskuja, jotka on koottu kuormakirjojen tiedoista.

Työ aloitetaan asiakastapaamisella, jossa määritellään asiakkaan tarpeita ja sovelluksen toivottuja perusominaisuuksia. Tästä tapaamisesta tuotetaan alustava vaatimusmäärittely. Asiakkaan kanssa myös hahmotellaan jo tulevan sovelluksen toimintaa piirtämällä kuvia, koska niin sovelluksen toimintaa on helpompi havainnollistaa.

Tapaamisen pohjalta luodaan myös ensimmäiset rautalankamallit sovelluksen keskeisimmistä näkymistä. Näiden karkeiden mallien tarkoitus on havainnollistaa sovellusta asiakkaalle ja saada hänet kertomaan mielipiteensä ja ehdottamaan omia parannuksiaan.

Yrityksen toimipisteessä Rengossa seuraavan vierailun aikana ja jälkeen kehitellään käyttäjätarinoita. Ne koskevat kuljettajien mobiilikäyttöliittymää ja toimiston työpöytäsovellusta. Käyttäjätarinoita varten voidaan haastatella kuljettajia. Myös kalustoon ja muuhun sovellusta koskevaan yrityksen materiaaliin tutustutaan tällä kerralla.

Käyttäjätarinat auttavat hahmottamaan sovelluksen todellista laajuutta ja työmäärää. Kokonaisuuden hahmottaminen on tässä projektissa erityisen tärkeää, sillä tulevaisuudessa sovelluskehityksen hinta- ja työmäärä-arvio pitäisi saada mahdollisimman tarkaksi.

Työ sisältää useita iterointikiertoja, joiden avulla sovellusta kehitetään asiakkaan haluamaan suuntaan. Tämä tarkoittaa käytännössä useita rautalankamalleja.

5 VAATIMUSMÄÄRITTELY

5.1 Tavoitteiden määrittely

Työn perimmäisinä tavoitteina on tehostaa yrityksen toimintaa ja vähentää työmäärää. Asiakkaan aika kuluu toimistossa erilaisten paperitöiden parissa, joista merkittävä osa on kuormakirjoihin liittyvää.

Kuormakirjojen arkistointi ja käsittely ovat yleisiä töitä toimistolla. Lisätyötä teettää eri työntekijöiden käsialan ja kirjoitusvirheiden tulkkaus. Kuormakirjat voivat myös hukkuu tai vahingoittua sekä unohtua jonnekin eri pituisiksi ajoiksi. Kuljettajat voivat tehdä tavallisuudesta poikkeavia merkintöjä, joiden merkitystä voi joutua kyselemään myöhemmin.

Kuljettajien sovelluksen halutaan tehostavan työntekoa ja yhdenmukaistavan kuormakirjojen laatimista. Sovellus muistaa ja osaa ennakoida esimerkiksi asiakkaiden ja työmaiden osoitteita sekä jo aiemmin kuljettujen matkojen pituuksia, jos sama väli ajetaan uudestaan. Sovelluksessa on myös korjauslista, johon voidaan rekisterinumeron perusteella kirjoittaa muihin vikoja ja korjauskohteita autoissa.

Tavoitteet:

- Kuormakirjojen hallinnan tehostaminen
- Toimistotöiden määrän vähentäminen
- Kuormakirjojen tuoton helpottaminen
- Kuormakirjojen yhdenmukaistaminen
- Tuntiseuranta
- Korjauslista
- Laskutus

5.2 Tarpeiden tunnistus

Asiakas tarvitsee työpöytäsovelluksen, joka on yhteydessä kuljettajien mobiililaitteisiin ja -sovellukseen kuorma-autoissa. Kuskien pitää pystyä tuottamaan ohjelman avulla vaatimusten mukaisia kuormakirjoja. Sovelluksen pitää pystyä saamaan kuljettajalta kaikki tarvitsemansa tiedot, jotta se voi luoda kuormakirjan. Kuljettajan täytyy pystyä korjaamaan virheitään, vaikka kuormakirja olisi jo lähetetty järjestelmään, myös pitkän ajan päästä.

Toimistosovelluksessa täytyy pystyä selaamaan kuormakirjoja, jotka on arkistoitu järjestelmään. Valtavaa kuormakirjamäärää tulee pystyä jäsentelemään esimerkiksi asiakkaan, kuljettajan tai päivämäärän mukaan.

Toimistosovelluksessa tulee olla tuntiseurantajärjestelmä. Tuntiseuranta-tietoja pitää pystyä jäsentelemään kuten kuormakirjojakin. Jäsentely tapahtuu samoilla periaatteilla esimerkiksi yhdelle työmaalle tehtyjen tai kuljettajan tietyllä aikavälillä tekemien työtuntien perusteella.

Tarpeet:

- Täydellisten kuormakirjojen tuotto
- Tuotettujen kuormakirjojen korjaus jälkeensä
- Kuormakirjojen arkistointi
- Kuormakirjojen selaus ja jäsentely
- Tuntiseurantajärjestelmän datan tuotto kuskeille ja tarkastelumahdollisuus toimistolle

5.3 Jalostetut vaatimukset

Aiemmista vaiheista jalostetaan kolmannessa vaiheessa lopulliset vaatimukset, jotka ovat suurilta osin vastauksia tarpeisiin. Seuraavaksi voidaan siirtyä tuottamaan rautalankamalleja, joissa kuvataan tapoja, joilla vaatimukset saadaan täytettyä.

Käyttäjätarinoissa osa vaatimuksista voidaan muuttaa tarinalliseen muotoon. Tarinamuotoisuus helpottaa vaatimusten taustalla olevien syiden ymmärtämistä.

Vaatimukset:

- Kuskien järjestelmällä täytyy pystyä tuottamaan täydellisiä kuormakirjoja
- Mahdolliset virheet täytyy pystyä korjaamaan myöhemmin
- Ohjelman täytyy täyttää mahdollisimman monia tietoja itsenäisesti
- Ohjelma tuottaa dataa yrityksen toiminnasta, kuten kuormien suuruuksista, ajomatkoista, kaluston korjaustarpeista ja työtunneista
- Toimistosovelluksella pitää pystyä tarkastelemaan kuormakirjoja
- Toimistosovelluksella voi jäsenellä kerättyjä tietoja
- Laskujen tarkastelu toimistosovelluksessa

6 KÄYTTÄJÄTARINAT

Käyttäjätarinat on tässä jaoteltu sovelluksen kahden eri käyttäjäkunnan mukaan Kuljettajiin ja Toimistoon. Kuljettajat käyttävät mobiilisovellusta, kun taas toimistolla on käytössään työpöytäsovellus laajemmilla toiminoilla.

6.1 Kuljettajat

Kuljettaja kirjautuu sovellukseen tabletillaan, joka on autossa. Ohjelma muistaa kuljettajan nimen ja osaa lisätä sen kuormakirjoihin itse.

Kuljettaja kirjoittaa päivän ensimmäisen asiakkaan sovellukseen. Sovellus osaa ehdottaa työmaan nimiä itse, joista kuljettaja voi valita haluamansa tai kirjoittaa sen manuaalisesti. Kuljettaja ilmoittaa työlaitteensa ja antaa sen mahdollisen rekisterinumeron. Sovellus kysyy tarkemmin, jos tarve vaatii, esimerkiksi mitä kuorma-autolla kuljetetaan ja onko peräkärriä.

Kuljettaja kertoo sovellukselle lähtöpisteen, kohteen, kuorman suuruuden, tavaran laadun. Sovellus ehdottaa automaattisesti kohteeksi työmaan osoitetta ja muistaa osoitteiden välisiä matkoja automaattisesti. Sitten kuljettaja painaa aloitusnappia, jotta sovellus aloittaisi ajanoton. Kohteeseen saavuttuaan kuski pysäyttää ajanoton painamalla nappia uudestaan. Tästä syntyy kuormakirja. Ajanotto on vain tuntiseurantaa varten, ei laskutukseen.

Kuljettaja haluaa nähdä, kuinka monta kuormaa hän vei työpäivän aikana. Kuljettaja menee kuormakirjojen yhteenvetosivulle, jossa näkyvät kaikki päivän aikana tehdyt kuormat ja yhteenveto tavaramääristä kullekin reitille ja laadulle.

Kuljettaja huomaa talvi-illan hämärtyneen. Tabletti häikäisee ja haittaa ajoa. Kuljettaja laittaa pimeämoodin päälle, jolloin sovelluksen tausta tummenee ja hohde vähenee.

6.2 Toimisto

Toimistossa ohjelman käyttäjä kirjautuu sisään. Hän haluaa tutkia viime viikon kuormakirjoja. Hän siirtyy Kuormakirjat-osioon, jossa hän lajittelee kuormakirjat ajan mukaan. Hän siirtyy edelliseen viikkoon ja etsii haluamansa kohdan. Kuormakirjan voi avata erikseen tai tarkastella sen tietoja suoraan yleisnäkyvästä.

Käyttäjä haluaa tietää, kuinka paljon Jaakko Virtanen teki työtunteja viime viikolla. Käyttäjä siirtyy tuntiseuranta-osioon, jossa hän lajittelee tiedot kuljettajien nimen mukaan. Seuraavaksi hän haluaa tietää, kuinka paljon

tunteja tehtiin Hongistontien työmaalle. Käyttäjä lajittelee tiedot työmaan mukaan ja etsii Hongistontien tiedot.

Käyttäjä haluaa tarkastella laskuja viime viikolta, sillä hän uskoo unohtaneensa yhden. Käyttäjä avaa sovelluksen ja siirtyy Laskut-osioon. Siellä käyttäjä lajittelee laskut laatimisajan mukaan ja etsii haluamansa laskun. Löydettyään oikean laskun käyttäjä vielä tulostaa sen paperille, jotta voi arkistoida sen.

7 VARHAISET KONSEPTIT

7.1 Lähtötilanne – kuormakirja

Sovelluksen pääasiallinen tarkoitus on luoda kuormakirjoja (Kuva 4). Kuormakirja on kirjallinen tosite, joka kertoo kuljetustapahtumasta. Kuormakirjasta tehdään yleensä kolme kappaletta kalkeeripaperin avulla, yksi menee asiakkaalle ja kaksi kuljetusyritykselle kirjanpitoa, laskutusta ja arkistointia varten.

Perinteisessä kuormakirjassa on omat ongelmansa. Kuljettajien käsiala vaihtelee ja suoranaisia kirjoitusvirheitäkin voi tulla. Toimiston päässä taas tekstin ja erikoismerkintöjen tulkkaminen aiheuttaa lisää virheitä. Sovelluksen tarkoitus on yhdenmukaistaa kuormakirjojen laatu tulkkauksvirheiden minimoimiseksi.

PVM/TYÖAIKA	TAUOT h	KULJETUSTEHTÄVÄ	NOUTOPAIKKA Nimi/kohte	LAJIKE mitä kuljetetaan	AJOMATKA km	LASTAUSKONE Nimi/kenen	KUORMAT lkm/kpl	MÄÄRÄ	YKSIKKÖ m ³ /m ² yms.	Ä-HINTA €
ALKOI 23.10.-15 klo: 7.00	0,5	Markkulahti - Perkkoo	maa	13-14	Ahola	7	91	m ³		
PÄÄTTYI 16.00		Markkulahti - Nurmi (Jalli)	kerä	4-5		4	52	m ³		
ALKOI 26.10.-15 klo: 7.00	0,5	Markkulahti - Perkkoo	maa	13-14	Ahola	7	91	m ³		
PÄÄTTYI 16.30		- u - - Nurmi (Jalli)	kerä	4-5		3	39	m ³		
ALKOI 27.10.-15 klo: 7.00	0,5	Markkulahti - Perkkoo	maa	13-14	Ahola	9	117	m ³		
PÄÄTTYI 16.30		- u - - Nurmi (Jalli)	kerä	4-5		1	13	m ³		
ALKOI 28.10.-15 klo: 7.00	0,5	Markkulahti - Perkkoo	maa	13-14	Ahola	9	117	m ³		
PÄÄTTYI 15.30		- u - - Nurmi (Jalli)	kerä	4-5		1	13	m ³		
ALKOI 29.10.-15 klo: 7.00	-	Markkulahti - Perkkoo	maa	13-14	Ahola	6	78	m ³		
PÄÄTTYI 12.00		- u - - Nurmi (Jalli)	kerä	4-5		1	13	m ³		
YHTEENSÄ €										
HUOMAUTUKSIA/POIKKEAMAT										

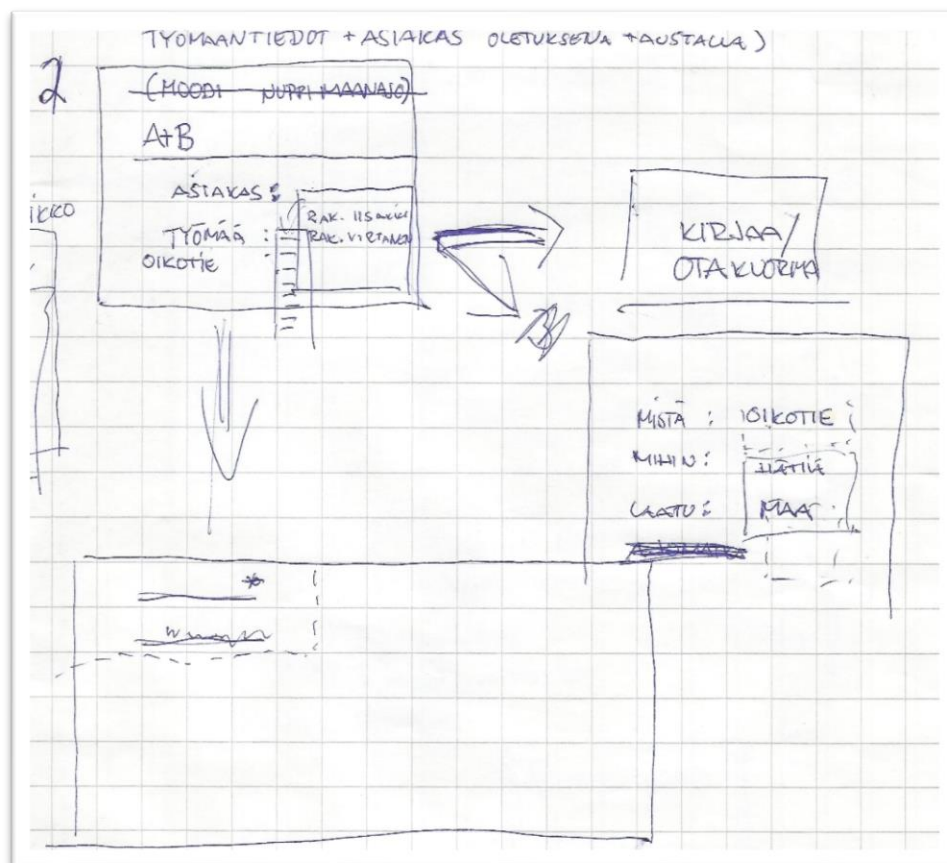
Kuva 4. Perinteinen kuormakirja. Skannaus.

7.2 Luonnokset

Ensimmäisen tapaamisen aikana luotiin suunnitelmia paperille. Paperissa on loistava käyttöliittymä ja se on tehokas konseptointiväline. Asiakas havainnollisti toiveitaan piirtämällä niistä kuvia suullisen selityksen tueksi (Kuva 5). Itsekin piirsinkin muutamia ehdotuksia, mutta tyydyin pääasiassa kommentoimaan asiakkaan havainnekuvia.

Jälkeenpäin katsottuna luonnokset voivat näyttää hyvinkin sekavilta, sillä ne on aina sidottu hetkeen ja piirtäjän kertomukseen ja ajatuksenjuoksuun. Piirroksista kannattaakin tehdä itselleen muistiinpanot mahdollisimman pian tapaamisen päätyttyä, kun asiat ovat vielä tuoreessa muistissa.

Luonnosten pohjalta piirrettiin ensimmäiset rautalankamallit sovelluksen keskeisistä näkymistä. Rautalankamalleja alettiin seuraavaksi iteroimaan eteenpäin.



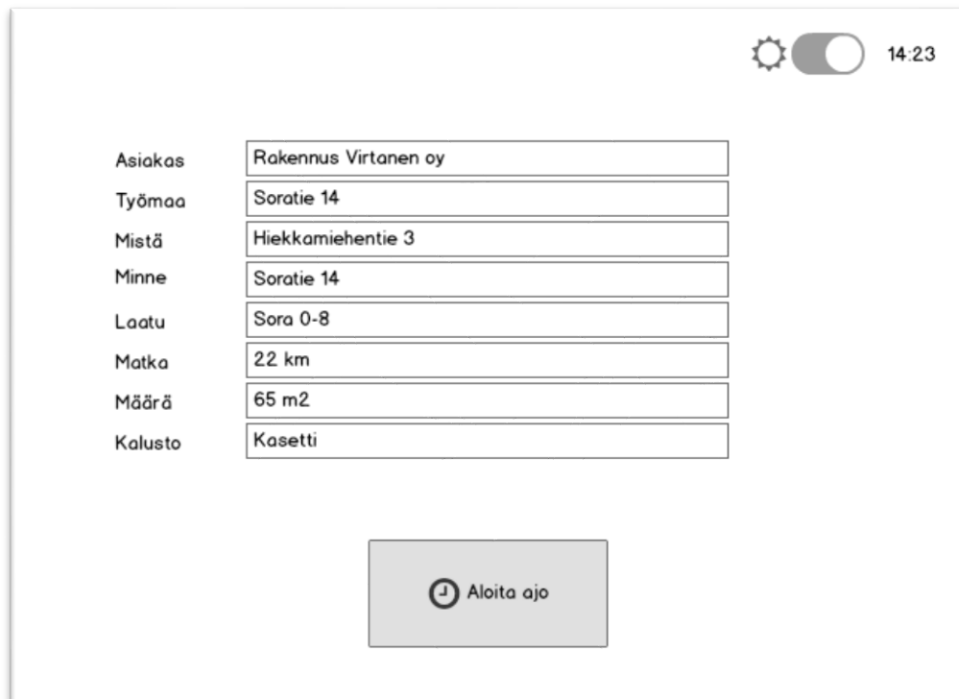
Kuva 5. Yksityiskohta asiakkaan kanssa tehdyistä sovellusluonnoksista. Kohta kuvaa Työn tiedot -kohdan varhaista vaihetta. Skannaus.

7.3 Ensimmäiset rautalankamallit

Työn alussa tutustuttiin Balsamiq-ohjelmaan ja piirrettiin muutamia kokeellisia kuvia lyhyen tapaamisen aikana tapahtuneen haastattelun pohjalta. Ensimmäisissä kuvissa tarkoituksena oli saada jonkinlainen visuaalinen ehdotus sovelluksesta, jota asiakas saisi kommentoida ja ohjata haluaansa suuntaan.

Yhdessä varhaisessa konseptissa (Kuva 6) kaikki mahdollinen on yritetty laittaa yhteen näkymään. Se ei tietenkään ole järkevää ja asiakas tyrmäsi tuotoksen suoriltaan. Kuvan pohjalta alettiin kuitenkin sijoitella tarvittavia

tietoja oikeaan syöttöjärjestykseen ja loogisiin osa-alueisiin. Tämä on ensimmäinen iterointikierron, joten siinä löytyy paljon muutettavaa eli tässä tapauksessa lähes kaikki. Iterointeja jatketaan, kunnes päästään riittävän lähelle täydellistä sovellusta.



The screenshot shows a mobile application interface with a form for data entry. The form consists of eight rows, each with a label on the left and a text input field on the right. The labels are: Asiakas, Työmaa, Mistä, Minne, Laatu, Matka, Määrä, and Kalusto. The input fields contain the following text: Rakennus Virtanen oy, Soratie 14, Hiekkamiehentie 3, Soratie 14, Sora 0-8, 22 km, 65 m2, and Kasetti. At the bottom center of the form is a button with a play icon and the text 'Aloita ajo'. In the top right corner, there is a settings gear icon, a toggle switch, and the time 14:23.

Asiakas	Rakennus Virtanen oy
Työmaa	Soratie 14
Mistä	Hiekkamiehentie 3
Minne	Soratie 14
Laatu	Sora 0-8
Matka	22 km
Määrä	65 m2
Kalusto	Kasetti

Aloita ajo

Kuva 6. Varhainen konseptikuva tablettinäkömästä.

8 RAUTALANKAMALLIT

Sovelluksen toiminnasta tuotetaan runsaasti rautalankamalleja. Niitä on helppo tuottaa ja hylätä nopeasti asiakkaan kanssa. Rautalankamallissa käyvät ilmi vain sovelluksen toiminnot ja sisällön sijoittelu, ei visuaalinen ilme.

Rautalankamalleissa punaisella merkityt osat ovat tekijän kommentteja. Yleensä kommentit ja huomiot hoidetaan suullisesti, mutta kirjallisessa ja kuvallisessa esityksessä täytyy käyttää selitetekstejä.

8.1 Mobiilisovellus

Tähän sovellukseen kirjaututaan ensimmäisenä sisälle (Kuva 7). Kuljettajat saavat omat tunnukset, joista sovellus myös saa nimen tuntikirjanpitoon ja kuormakirjoihin.



Kuva 7. Mobiilisovelluksen sisäänkirjautuminen.

Seuraavaksi annetaan työn tiedot (Kuva 8). Tämä kohta kattaa yhden asiakkaan, jolle tavallisesti tulee useita kuormia päivässä. Kuljettaja antaa asiakkaan nimen ja sovellus osaa jo ennakoida tietokannoistaan tai aiemmin syötettyjen tietojen perusteella, mitä asiakasta kuljettaja voisi tarkoittaa.

Asiakkailla voi olla useita työmaita, joihin viitataan usein katuosoitteella. Tässäkin sovellus muistaa, mitä työmaita asiakkailla on aktiivisena. Seuraavaksi valitaan laite. Valittu työlaite määrää, kumpi seuraavista kentistä tulee näkyviin. Rekisterinumeroisilla laitteilla, eli pääasiassa kuorma-autoilla, tulee näkyviin kenttä, jossa pyydetään rekisterinumeroa. Rekisterinume-

ron perusteella ohjelma tietää, mitä autoa käytetään, eli kuinka suuri kapasiteetti on käytössä. Erilaiset kaivinkoneet ja muut maan- ja tienrakennuslaitteet menevät rekisteröimättömien kategoriaan.

Työn tiedot

Ennakoiva tekstinsyöttö

Asiakas	Häm Hämeen sora oy Kanta-Hämeen kaivinkoneurakointi Tmi Mauno Hämmäläinen
Työmaa	Puolukkatie 14 Savipellontie 45 Karjalankuja 97 Loppuunajetuntie 3
Työlaite	Kuorma-auto Tiehöylä Lumioura Kaivinkone
Rekisterinumero	BRG-338
Laite	Kaivinkone 2 Kaivinkone 1 Kaivinkone 3 Bobcat

OK

Tulee näkyviin, jos valitaan laite jolla on rekisterinumero

Ehdotukset asiakkaan perusteella laitteen muistista tai tietokannasta

Tulee näkyviin, jos valitaan laite jolla ei ole rekisterinumeroa

Kuva 8. Työn tiedot -osio, joka tulee heti kirjautumisen jälkeen. Punaisella katkoviivalla merkityt alueet tulevat näkyviin vain tietyissä tapauksissa.

Jos edellisessä kohdassa valittiin kuorma-auto, jolle on annettu rekisterinumero, tulee seuraavaksi moodi (Kuva 9). Siinä valitaan kuljetettavan aikneksen laatu ja ilmoitetaan mahdollinen kasetti eli peräkärri.



Kuva 9. Moodin valinta, jossa määritellään kuljetettava aines ja peräkärri.

Seuraavassa vaiheessa (Kuva 10) kysytään aloitusaikaa aiemmin valitulla työmaalla, mutta vain ensimmäisellä kerralla kullekin asiakkaalle. Työn aloitusaika on erillinen kunkin kuormakirjan aloitusajasta. Kuorman otto työmaalla voi joutua odottamaan kauankin, jolloin työ saattaa mennä tunti hinnoittelulle. Aloitusaika tulee pystyä valitsemaan manuaalisesti, sillä kuljettaja voi haluta kirjoittaa tiedot jo aamukahvilla valmiiksi ja ilmoittaa ajan vasta kun pääsee työmaalle.

Seuraavaksi valitaan lähtöpaikka, jossa kuorma otetaan kyytiin ja paikka jossa se puretaan. Moodissa aiemmin valittua kuljetetun tavaran laatua voidaan tarkentaa, tässä esimerkissä on eri karkeusasteisia soralaatuja. Tavaran laatu ja käytettävä kuljetuskalusto määräävät sen, käytetäänkö kuormayksikkönä tilavuutta vai painoa. Sovellus osaa ehdottaa rekisterinumeron perusteella kunkin kuorma-auton oletustilavuutta.

Viimeisessä kohdassa valitaan kohteiden välinen etäisyys. Tämän voi määrittää myös juuri ennen purkua, jos ei tiedä kohteiden välistä etäisyyttä ennakoon. Sovellus muistaa aiemmin kuljettujen matkojen pituuksia ja osaa ehdottaa niitä automaattisesti, kun lähtö- ja purkupaikka on ilmoitettu.

Kuormakirjat

Kysytään vain ensimmäisellä kerralla

Työn aloitusaika	
4	05
5	00
6	01
7	02
8	03
9	04
10	05

Lähtöpaikka

Purkupaikka

Laatu

Määrä tai paino valikoituu kaluston ja tavarun laadun mukaan

Määrä/paino

Sovellus muistaa aiemmin kuljettujen välien pituuksia

Matka km

1-2
2-3
3-4
4-5
5-6
6-7
8-9

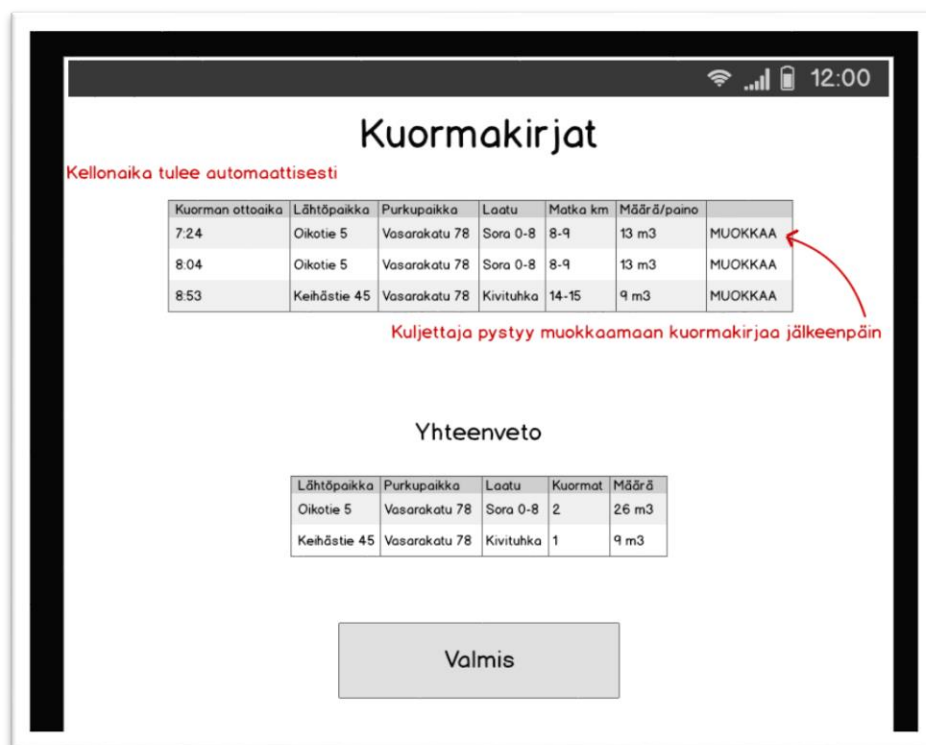
Pura
Aloita

} "Aloita" vaihtuu
"Puraksi"

Kuva 10. Kuormakirjanäkymä. Painike vaihtuu kuorman oton ja aloituksen jälkeen purkamisnapiksi.

Samalle asiakkaalle viedyistä kuormista muodostetaan yhteenveto (Kuva 11). Siitä käyvät ilmi osoitteet, kuormat, matkat ja tavaramäärät. Asiakkaalle tuotetuista kuormakirjoista muodostuu yhteenveto, josta niitä voi vielä muokata tarvittaessa. Virheelliset kuormakirjat voi myös poistaa kokonaan.

Kun kuljettaja painaa *valmis*-nappia, hän kuittaa kaikki listassa olevat kuormakirjat valmiiksi. Sovellus siirtyy takaisin *työn tietoihin*, jossa voidaan valita uusi asiakas ja työmaa. Kuljettaja voi tehdä kuormakirjoja vain yhdelle asiakkaalle kerrallaan.



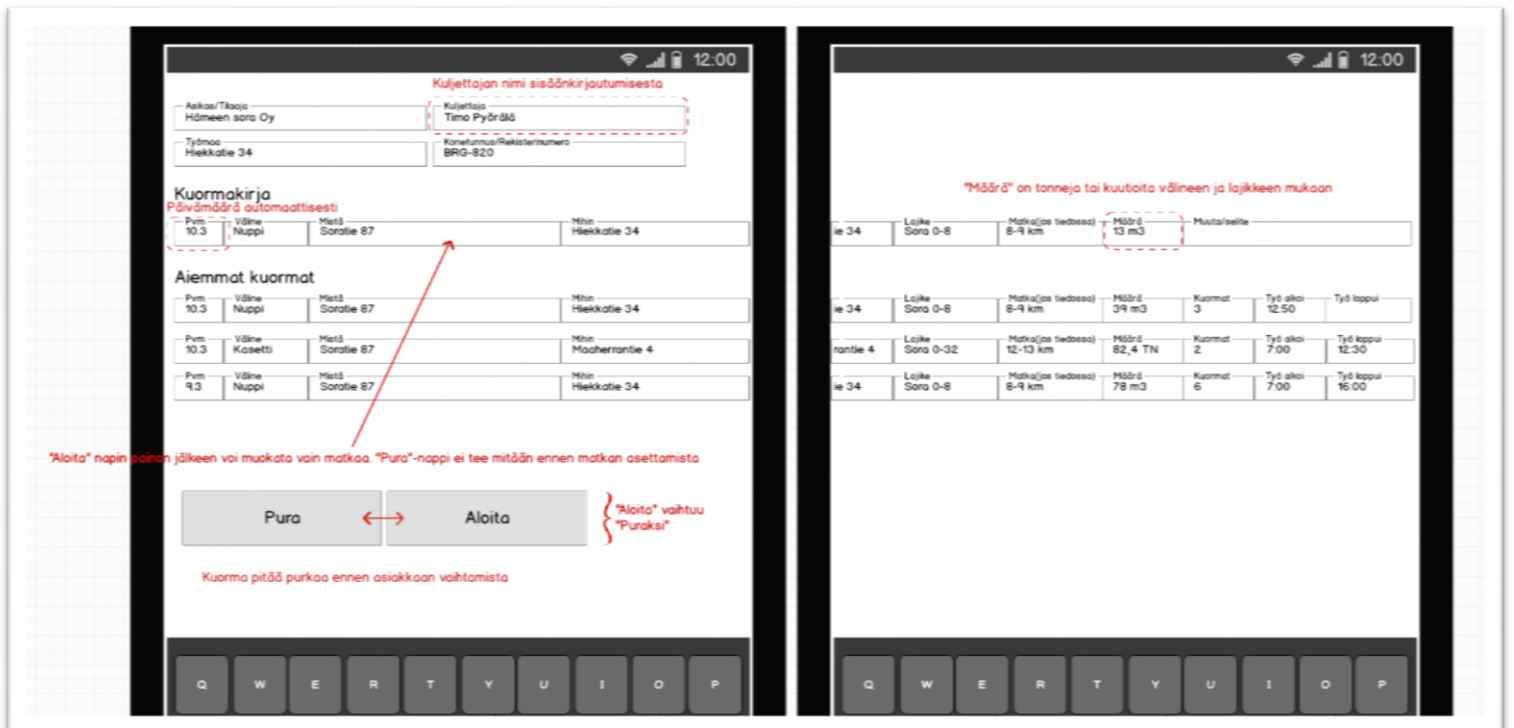
Kuva 11. Yhteenvetonäkymä asiakkaalle viedyistä kuormista.

8.2 Vaihtoehtoinen mobiilisovelluskonsepti

Tilaaaja halusi myös ylläesiteltujen lisäksi vaihtoehtoisen sovelluskuvan (Kuva 12). Siinä pohjana olisi paperinen kuormakirja ja sovellus imitoisi sen toimintaa. Siinä monet ominaisuudet on sijoitettu yhteen näkymään, joka on niin leveä, että se ei mahdu yhteen ruudulliseen. Yrityksen paperinen kuormakirjakin on vaakatasossa olevan A4-kokoisen kalkeeripaperin muodossa.

Tällainen sovellus olisi tilaajan mielestä kuljettajille helpommin lähestyttävä normaalin kuormakirjan ulkoasun johdosta. Kuvat muistuttavat enemmän täytettävää lomaketta, mutta olisivat varmasti toimivat. Joitain elementtejä voidaan ottaa suoraan aiemmista konsepteista.

Kenttiä on paljon, mutta kirjoittaminen toteutettaisiin tarkentamalla haluttuja alueita automaattisesti tai käyttämällä tablettien normaaleja tarkennusominaisuuksia. Myös sovelluksen teko oletuksena vaakatasossa käytettäväksi voisi auttaa tilaongelmassa.



Kuva 12. Vaihtehtoinen sovelluskuva pystysuuntaisesti kahdelle näyttönäkymälle levitettyä.

8.3 Toimistosovellus

Työssä keskityttiin mobiilisovelluksen suunnitteluun, mutta asiakas halusi muutaman kuvan myös työpöytäsovelluksen mahdollisesta ulkoasusta. Sovellus olisi normaali Windows-ympäristössä toimiva ohjelma, joka on yhteydessä samaan tietokantaan, kuin mobiilisovelluksetkin. Tietokanta voi sijaita pilvipalvelussa tai perinteisillä alustoilla.

Kuljettajien sovelluksen tuottamaa dataa seurataan, jäsenellään ja hallitaan toimistosovelluksella. Sieltä löytyvät kuljettajien tuottamat kuormakirjat, asiakkaiden tiedot, laskutus ja tuntiseuranta. Ohjelmalla pitää pystyä seuraamaan kuormakirjojen lisäksi myös muita yrityksen tekemiä töitä, kuten lumi- ja tietöitä.

Kuvissa näkyvät välilehdet *kartta* ja *ilmoitustaulu* karsittiin lopullisista kuvista pois, mutta jätettiin tähän havainnollistamaan ominaisuuksien karsimista. Ne todettiin ainakin alkuun turhiksi ominaisuuksiksi, jotka vain kallistaisivat ja monimutkaistaisivat sovellusta. Sovelluksen myöhemmissä versioissa, jolloin se on jo kuljettajille tuttu, voitaisiin nämä ominaisuudet ja paljon muitakin liittää sovellukseen tarpeen mukaan.

Kuormakirjoja tulee suuri määrä ja niitä pitääkin pystyä jäsentelemään (Kuva 13). Hyviä lajittelutekijöitä ovat tässä tapauksessa esimerkiksi aika, kuljettaja, asiakas ja kuorma-auton rekisterinumero.

Ainakin alkuun tilaaja haluaa vielä tehdä laskunsa käsin. Sovelluksen laskutusosiossa näytettäisiin vain laskuihin tarvittavia tietoja kuormakirjoista, joiden avulla toiseen sovellukseen syötetään tarvittavat tiedot. Tulevaisuudessa laskujen tuotonkin voisi automatisoida.

Kuljetus Viikkinen Oy

Kuormakirjat Muu työ Tuntiseuranta Laskutus Kartta Ilmoitustaulu

Lajittelua voi rajata useampien tekijöiden avulla

Lajittelu Asikas Työmaa
 Kuskki Kuskki
 Aika-Kaksi päivää Aika-Kaksi päivää
 Aika-Viikko Aika-Viikko
 Aika-Kuukausi Aika-Kuukausi
 Aika-Vuosi Aika-Vuosi
 Aika Aika
 Työn laatu Työn laatu
 Auto-Rekisterinumero Auto-Rekisterinumero
 Matka Matka
 Aika Aika

Monosen rakennusliike

Hämeen sora oy

T.mti Jaakko Hiekkala	Työmaa	pvm
Avaa kuormakirja	Menninkäisentie 87	2.2.2017
Avaa kuormakirja	Menninkäisentie 87	2.2.2017
Avaa kuormakirja	Menninkäisentie 87	3.2.2017

Linkistä pääsee tarkastelemaan kuormakirjan sisältöä kokonaisuudessaan

Kuva 13. Mahdollinen kuormakirjanäkymä toimistosovelluksesta.

Sovelluksen on tarkoitus tuottaa tuntiseurantatietoja (Kuva 14), jotka tulevat kuljettajan sovelluksesta. Kuljettaja kirjaa työn aloitusajaksi sen hetken, kun hän saapuu asiakkaan työmaalle. Työn aloitusajalla on merkitystä, sillä se voi erota merkittävästi ensimmäisen kuorman lastauksesta. Tämä voi johtua esimerkiksi työmaan henkilöstön hitaasta aamutoiminnasta tai laitevioloista.

Sovelluksen tuottamaa aikadataa ei ole tarkoitettu vain työntekijöiden työtehojen seurantaan, vaan pikemminkin eri reittien ja työmaiden kannattavuuden selvittämiseen mahdollisimman nopeasti. Jos vaikkapa reitti on hyvin vaikeakulkuinen tai lastaava työmaa tehoton, voidaan asiaan puuttua nopeasti tappioiden vähentämiseksi.

Työaika	Pvm	Työmaa
2h	3.2.2017	Niittykatu 5
6h	3.2.2017	Raksaniementie 56

Kuva 14. Konseptikuva tuntiseurannasta työpöytäsovelluksessa.

9 YHTEENVETO

Opinnäytetyön tarkoituksena oli perehtyä käyttöliittymäsuunnitteluun ja tuottaa tilaajalle konseptikuvia tulevasta sovelluksesta. Sovelluksen suunnittelussa oli punaisena lankana helppokäyttöisyys. Kävi ilmi, että helppokäyttöisyys on subjektiivinen käsite. Ihmisillä on erilaiset käsitykset siitä, mikä on helppoa käyttää ja mikä ei, pohjautuen omiin kokemuksiin, tietoihin ja taitoihin. Helppokäyttöisyystavoite toteutui kuitenkin tyydyttävästi.

Työssä tuotettiin tilaajalle paljon kuvia sovelluksesta, joiden avulla hän voi tehdä tarjouspyyntöjä sovelluksia tekeville yrityksille. Sovelluksen lisäksi pitää hankkia palvelimet ja muut oheisjärjestelmät. Tilaajan tarpeiden ja toiveiden kartoitus oli myös tärkeä osa työtä. Usein asiakkaat eivät tarkkaan tiedä, mitä haluavat uudelta sovellukseltaan ja mitä pitäisi ottaa huomioon sellaista suunniteltaessa.

Suunnittelumenetelmien tutkiminen oli hyvä asia opiskelun ja kertauksen lisäksi myös siksi, että se auttoi oivaltamaan, että on olemassa monia erilaisia suunnittelutekniikoita. Ne ovat osittain päällekkäisiä, osin toisiaan tukevia. Kynä ja paperikin ovat oivallisia konseptointityökaluja, vaikka ei osaisikaan piirtää erityisen hyvin.

Myös maansiirtoyrityksen toimintaa tuli työssä käytyä läpi pintaa syvemältä. Sekään ei ole yksinkertaista työtä, varsinkaan jos sen yrittää puristaa sovelluksen sisälle ulkopuolisena.

LÄHTEET

- Android (n.d). Icon design guidelines. Haettu 12.2.2017
https://developer.android.com/guide/practices/ui_guidelines/icon_design.html
- Bridging the gap (2017). How to write a use case. Haettu 28.2.2017
<http://www.bridging-the-gap.com/what-is-a-use-case/>
- Helsingin yliopiston tietojenkäsittelytieteen laitos (n.d) , ohjelmistotutustannan luento 3. Haettu 24.1.2017
<https://www.cs.helsinki.fi/group/java/k12-ohtu/luento3.pdf>
- Immonen, J. (2002). Graafiset käyttöliittymät. Luentomoniste, Joensuun yliopisto. Haettu 12.2.2017
http://cs.joensuu.fi/~jimmonen/gkl_moniste/gkl_v202.html
- Krug, S. (2006). *Älä pakota minua ajattelemaan!* 2. laitos. Jyväskylä: Gummerus kirjapaino Oy.
- Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. (2002). *Käytävyyden psykologia*. Helsinki: Edita Oyj.
- Sinkkonen, I., Nuutila, E. & Törmä, S. (2009). *Helppokäyttöisen verkkopalvelun suunnittelu*. Hämeenlinna: Kariston kirjapaino Oy.
- Stellman-Greene (2009). Requirements 101: User Stories vs. Use Cases. Haettu 24.1.2017
<https://www.stellman-greene.com/2009/05/03/requirements-101-user-stories-vs-use-cases/>
- Tieke (2005). Vaatimusmäärittely. Haettu 24.1.2017
<http://www.tieke.fi/pages/viewpage.action?pageId=3441242>
- Vibor Cipan (2010). User interfaces for beginners, intermediates or experts? Haettu 23.2.2017
<http://www.uxpassion.com/blog/user-interface-design-beginners-intermediates-experts/>