

SÄHKÖINEN NUOHOUKORTISTO PIENYRITYKSELLE

LAHDEN AMMATTIKORKEAKOULU
Tradenomi (AMK)
Tietojenkäsittely
Kevät 2019
Tuomas Honkanen

Tiivistelmä

Tekijä(t) Honkanen, Tuomas	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 34	Valmistumisaika Kevät 2019
Työn nimi Sähköinen nuohouskortisto pienyritykselle		
Tutkinto Tradenomi (AMK)		
Tiivistelmä <p>Toiminnallisen opinnäytetyön tarkoituksena oli toteuttaa sähköinen nuohouskortisto Nuohouspalvelu M. Honkaselle. Toimeksiantaja tarjoaa asiakkailleen nuohousalan palveluita Etelä-Karjalan alueella. Yrityksellä on aiemmin ollut käytössään käsin kirjoitettu, täysin paperinen nuohouskortisto. Sähköistämiprojektilla pyrittiinkin tehostamaan yrityksen toimintaa, sillä paperisen kortiston ylläpitäminen on työlästä sekä tilaa ja aikaa vievää.</p> <p>Teoriaosassa tutkittiin, mitä pienyrityksen on syytä huomioida sähköiseen järjestelmään siirtymisessä. Työssä perehdyttiin sähköisen ohjelmaprojektin eri vaiheisiin sekä määriteltiin, suunniteltiin ja toteutettiin toimeksiantajan tarpeet täyttävä sähköinen nuohouskortisto. Tarvittava taustatieto kerättiin alan kirjallisuudesta, internetistä sekä haastattelemalla toimeksiantajaa.</p> <p>Toimeksiantajan vaatimusten ja toiveiden pohjalta sähköinen nuohouskortisto päädyttiin toteuttamaan verkkoselainpohjaisena sovelluksena. Sovelluksen toteutuksessa hyödynnettiin HTML-, CSS-, ja PHP-kieliä. Yrityksen omistama pöytäkone muutettiin Linux-palvelinkäyttöön sopivaksi, ja ohjelma sekä sen tietokanta siirrettiin kyseiselle laitteelle.</p> <p>Kokonaisuutena sähköisen nuohouskortiston toteutus onnistui hyvin, ja lopputuote täyttää suurimman osan asiakkaan kanssa sovituista ratkaisuista. Käyttöön otettu kortisto helpottaa ja nopeuttaa yrityksen kortistotietojen käsittelyä huomattavasti. Tutkimus ja työn toteutus osoittivat, että sähköistämiprojektissa on useita eri vaiheita. Lisäksi pienyrityksen täytyy valita sopiva toteutusvaihtoehto ja huomioida tietoturva. Projektin onnistuminen kaikkia osapuolia tyydyttävällä tavalla vaatii siis asiantuntemusta ja tiivistä yhteistyötä toimeksiantajan kanssa.</p>		
Asiasanat pienyritys, verkkoselainpohjainen sovellus, sähköinen nuohouskortisto		

Abstract

Author(s) Honkanen, Tuomas	Type of publication Bachelor's thesis	Published Spring 2019
	Number of pages 34	
Title of publication An Electronic Index Card System for a Chimney Sweeping Company		
Name of Degree Bachelor of Business Administration		
Abstract <p>The aim of this functional thesis was to produce an electronic index card system for Nuohouspalvelu M. Honkanen. The client company offers chimney sweeping services to its customers in the South Karelia region. The company has used hand written paper index cards, but such a system takes a lot of effort and time to maintain. Therefore, by transferring over to an electronic solution the company aims to improve its performance.</p> <p>The theoretical section of the thesis studies the main aspects small businesses should to take into account when switching to an electronic system. The thesis goes through the different phases of a software project, resulting in an application, which fulfills the customer's needs. Interviews, literature and the internet were used to gather the required source material for the study.</p> <p>The application was developed based on the needs assessment. HTML, CSS and PHP languages were used in the making of the implementation. The application and its database run on the client's computer after a Linux based server environment was first installed on it.</p> <p>As a whole, the development of the application went well. The end product fulfills most of the client's needs, and it makes saving and editing the gathered chimney sweeping information faster. The research and the implementation demonstrate the many steps of a software development project. In addition it shows that small businesses need to pay close attention to the chosen implementation methods, and the security of the hardware and the software. Therefore, for a project to succeed in a way that satisfies every party involved, expertise and close cooperation with the client are required.</p>		
Keywords Small business, WEB-development, electronic index cards, chimney sweeping		

SISÄLLYS

1	JOHDANTO	1
1.1	Tutkimuksen tausta ja rajaus	1
1.2	Tutkimuksen tavoitteet	1
1.3	Aiemmat tutkimukset	2
1.4	Tutkimus-, tiedonkeruu- ja analyysimenetelmät	2
2	PIENYRITYS SÄHKÖISEN TIETOJÄRJESTELMÄN PIIRIIN.....	5
2.1	Järjestelmän valinta	5
2.2	Tietojen käsittely	5
2.3	Tietoturva	6
3	NUOHOUSKORTISTO-OHJELMAN MÄÄRITTELY	8
3.1	Nykytilanteen kartoittaminen	8
3.2	Toteutusvaihtoehdon valinta	8
3.3	Vaatimukset	9
3.4	Käyttötapaukset	10
4	SÄHKÖISEN NUOHOUSKORTISTON SUUNNITTELU	12
4.1	Sovelluksen ulkoasu	12
4.2	Palvelinympäristön suunnittelu	12
4.3	Tietokanta ja siihen siirrettävät tiedot	13
5	SÄHKÖISEN NUOHOUSKORTISTON TOTEUTUS	14
5.1	Projektinhallintamalli	14
5.2	Laitteet ja ohjelmistot	16
5.3	Linux-pohjaisen palvelimen käyttöönotto	18
5.4	Tietokanta	18
5.5	Sovellus	18
5.6	Testaus palvelinympäristössä	25
5.7	Käyttöönotto	25
5.8	Varmuuskopiointi ja tietoturva	26
6	YHTEENVETO	27
	LÄHTEET	29
	LIITTEET	31

1 JOHDANTO

1.1 Tutkimuksen tausta ja rajaus

Opinnäytetyön toimeksiantaja on vuonna 2010 perustettu nuohousalan yritys Nuohouspalvelu M. Honkanen. Yrityksen toimintamuotona on yksityinen elinkeinonharjoittaja, eikä yrityksellä ole tällä hetkellä yrittäjän lisäksi muita työntekijöitä. Yrityksen päätoiminta-alueena on Etelä-Karjala, ja sen tarjoamiin palveluihin kuuluvat tulisijojen ja savuhormien nuohous sekä lämmityskattiloiden puhdistukset.

Toimeksiantaja on aiemmin pitänyt asiakastietoja sisältävää nuohouskortistoa paperisessa muodossa. Nuohouskorttiin on kerätty vain nuohoustyön suorittamisen kannalta oleelliset tiedot. Toimeksiantaja on jo pitkään harkinnut nuohouskortiston siirtämistä sähköiseen muotoon, sillä korttien kirjoittaminen käsin on työlästä ja aikaa vievää. Sähköistäminen tehostaisi myös yrityksen toimintaa ja helpottaisi merkittävästi uuden tiedon syöttämistä, olemassa olevan tiedon hakemista, muokkaamista sekä arkistointia.

Nuohousalalla ei ole Suomessa yhtä paljon toimijoita kuin monilla muilla palvelualoilla, kuten esimerkiksi parturi- ja kampaamoalalla. Markkinoilla onkin valmiita juuri nuohousalalle soveltuvia sähköisiä kortistoratkaisuja varsin vähän. Toimeksiantaja on jo aikaisemmin perehtynyt valmISRatkaisuihin, mutta ne eivät vastaa täysin yrityksen tarpeita.

Opinnäytetyö sähköisen nuohouskortiston toteuttamisesta perustuu työelämään, ja se on suoritettu toimeksiantajan tämänhetkisten tarpeiden täyttämiseksi. Tutkitut ja käytetyt ohjelma- ja laitteistoratkaisut ovat siis rajoittuneet toimeksiantajan toiveiden ja vaatimusten mukaan. Tutkimusosa rajoittui pienyrityksiin, jotta siitä olisi hyötyä myös kohdeyritykselle. Opinnäytetyön ulkopuolelle on rajattu myös oikeiden asiakastietojen käsitteleminen ja syöttäminen valmiiseen järjestelmään, joten vaitiolovelvollisuussopimusta ei tarvita.

1.2 Tutkimuksen tavoitteet

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa toimeksiantajan vaatimukset sekä toiveet täyttävä sähköinen nuohouskortisto. Sen tulee vastata tietosisällöltään pääosin jo olemassa olevaa paperista kortistoa mutta sähköistämisen myötä tehdä asiakastietojen käsittelystä helpompaa ja tehokkaampaa. Kortiston tulee myös olla tarpeeksi helppokäyttöinen ja selkeä, jotta yrityksen omistaja osaa käyttää sitä ilman merkittävää koulutuksen tarvetta. Tietoturvallisuuskin on syytä huomioida sähköisen kortiston toteutuksessa.

Sähköiseen tietojärjestelmään siirtyminen on yritykselle haastava prosessi. Tutkimuksen toisena tavoitteena onkin selvittää tärkeimpiä seikkoja, joita pienyrityksen tulee huomioida sähköisen asiakastietojärjestelmän valinnassa ja siihen siirtymisessä.

1.3 Aiemmat tutkimukset

Nuohouskortiston sähköistämiseen liittyviä aiempia tutkimuksia ei löytynyt. Muunlaisia sähköisiä tietojärjestelmiä käsitteleviä tutkimuksia on kuitenkin tehty. Esimerkkinä tällaisesta tutkimuksesta on Tuiskun ja Niemelän YAMK-opinnäytetyö, jossa tutkitaan sähköisen asiakastietojärjestelmän hankintaa terveysalan yritykselle. Työssä kartoitetaan tarkasti järjestelmän ja asiakkaan vaatimukset, joiden pohjalta yritys sai neljä kriteerit täyttävää tarjousta asiakastietojärjestelmistä. Tuiskun ja Niemelän tutkimuksesta ilmenee, että oikean järjestelmän valinta ja hankinta on suuri projekti erityisesti pienelle yritykselle. Onnistuneella tarpeiden kartoittamisella voidaan kuitenkin helpottaa sopivan järjestelmän hankintaa. (Tuisku & Niemelä 2017.) Terhon (2016) Pro gradu -tutkielma puolestaan käsittelee kulunvalvonta- ja asiakastietojärjestelmän hankintaa liikuntakeskukselle. Työssä tutkitaan vaatimusten määrittelyä ja tietojärjestelmän hankintaa pk-yrityksen kannalta. Terho perehtyy myös liikuntatoimialaan syvällisemmin, koska alan ymmärtäminen helpottaa järjestelmävalinnan suorittamista.

Opinnäytteiden lisäksi aiheeseen liittyen on tehty myös muita tutkimuksia. Pienten ja keskisuurten yritysten digitalisaatiota on tutkittu Elisan ja Suomen Yrittäjien tutkimuksessa. Tutkimuksesta käy ilmi, että kyselyyn vastanneista 559 yrittäjästä vain noin puolet hyödyntää digitaalisia liiketoimintatyökaluja. Työkaluja käyttöön ottaneista yrityksistä kuitenkin suurin osa koki digitalisaation tehostavan yrityksen toimintaa joko selvästi tai jonkin verran. (Elisa 2015.) Valtion teknillisen tutkimuskeskuksen julkaisu Toiminnanohjausjärjestelmän käyttöönotto PK-yrityksessä tarkastelee tietojärjestelmähankkeita laaja-alaisesti. Julkaisu sisältää kahdeksan artikkelia, joista ensimmäinen käsittelee tietojärjestelmien käyttöönoton kaikkia vaiheita. Toinen, neljäs, viides ja kuudes artikkeli puolestaan keskittyvät tarkastelemaan aihealuetta pk-yrityksien kannalta. (Kettunen & Simons 2001.)

1.4 Tutkimus-, tiedonkeruu- ja analyysimenetelmät

Toiminnalliset opinnäytetyöt ovat käytäntöä ja teoriaa yhdistelevä opinnäytetyötyyppi. Tällaisissa opinnäytetöissä aihealueet eivät ole tarkkaan rajattuja, mutta kaikille on yhteistä työelämäläheisyys ja jonkin tuotteen tai palvelun toteuttaminen. (Vilkkä & Airaksinen 2003, 9-10.) Toimintatutkimus on tutkimusmenetelmä, jossa tarkastellaan yrityksen tai organisaation nykyisiä toimintatapoja sekä käsityksiä. Nykytilanteen ymmärtämisellä pyritään ke-

hittämään toimintaa järjestelmällisesti paremmaksi. (Mcniff & Whitehead 2000, 207.) Nuohouskortistoprojekti on siis tyypiltään toiminnallinen opinnäyte, koska sen lopputuloksena on sähköinen nuohouskortisto toimeksiantajalle. Tutkimusmenetelmänä on puolestaan toimintatutkimus, sillä siinä perehdytään vanhaan paperiseen kortistoon, ja sen pohjalta päädytään paremmin toimivaan sähköiseen ratkaisuun.

Laadullista eli kvalitatiivista tutkimusta käytetään, kun työn onnistumisen kannalta on tärkeää ymmärtää toteuttamisen eri vaiheet syvällisesti ja laaja-alaisesti. Laadullinen tutkimusmenetelmä soveltuu myös projekteihin, joissa kohderyhmän ideat ja ehdotukset huomioidaan. (Vilkkä & Airaksinen 2003, 63.) Sähköisen nuohouskortiston toteuttamiseen onkin käytetty laadullista tutkimusmenetelmää. Se soveltuu hyvin juuri tämänkaltaiseen opinnäytetyöhön, jossa tiivis yhteistyö on keskeisessä roolissa ja työ tulee toimeksiantajan tarpeeseen.

Laadullisessa tutkimuksessa tiedon keräämistävän valinnassa on ratkaisevaa, miten sen halutaan edesauttavien omia tutkimustavoitteita. Teemahaastattelu on haastattelutyyppi, joka mahdollistaa suuntaa-antavan tiedon keräämisen vapaamuotoisesti, esimerkiksi kasvotusten haastatteleamalla tiettyyn teemaan liittyen. Se on myös paljon joustavampi tapa kerätä tietoa kuin tarkkarakenteinen haastattelulomake, jossa kysytään tarkasti määritellyt kysymykset tietyssä järjestyksessä. Hyvin toteutetulla haastattelulla saadaan laadukasta sekä opinnäytetyön toteuttamisen kannalta tärkeää tietoa. (Vilkkä & Airaksinen 2003, 63-64.) Nuohouskortiston sähköistämiprojektissa päädyttiinkin käyttämään tiedonkeruumenetelmänä teemahaastattelua. Haastateltava henkilö on yrityksen omistaja, ja haastattelun teemoja ovat nuohouskortiston nykytilanne, sähköisen kortiston tarpeet ja vaatimukset sekä yrityksen käytettävissä olevat resurssit. Avoimessa haastattelu- ja suunnittelutilaisuudessa toimeksiantajan tarpeet on mahdollista saada hyvin esille, sillä haastateltavalla henkilöllä ei ole kattavaa tietoteknistä taustaa. Toisaalta haastateltavalla on myös sellaista työn suorittamisen kannalta oleellista nuohousalan ammattitietoa, joka opinnäytetyön tekijältä puuttuu. Haastattelun aikana toimeksiantajan kanssa piirretään ja ideoidaan erilaisia kevyitä luonnoksia ohjelman ulkoasusta ja toiminnasta.

Laadullisissa tutkimuksissa yleisesti käytetty analyysimenetelmä on sisällönanalyysi. Siinä aineiston sisältö käydään läpi, ja merkityksellinen tieto lajitellaan tutkimuksen tavoitteiden mukaisesti, esimerkiksi teemoittain. Sisällönanalyysi voi olla aineisto- tai teorialähtöistä. Aineistolähtöisessä analyysissä pääpaino on täysin aineistossa, kun taas teorialähtöinen analyysi perustuu johonkin aikaisempaan viitekehukseen, kuten olemassa olevaan teoriaan. (Tuomi & Sarajarvi 2009, 91-97.) Sähköisen nuohouskortiston teemahaastattelun

analyysimenetelmänä on käytetty aineistolähtöistä sisällönanalyysiä, jossa yksilöhaastattelussa saatua aineistoa on helppo analysoida valmiiden teemojen pohjalta.

2 PIENYRITYS SÄHKÖISEN TIETOJÄRJESTELMÄN PIIRIIN

2.1 Järjestelmän valinta

Sähköiseen tietojärjestelmään siirtyminen on aina iso askel yrityksen toiminnalle. Sähköiset järjestelmät voivat monilta osin helpottaa yrityksen toimintaa, mutta toteutusratkaisujen valinnassa, niiden käyttöönotossa sekä käytössä on omat haasteensa. Yrityksien täytyy olla tarkkana valitessaan käyttötarpeisiinsa sopivaa sähköistä asiakastietojärjestelmää. Nykyään kotimaisten ja ulkomaisten tuotteiden sekä palveluiden tarjoajia on runsas määrä. Suurien ohjelmistotalojen ratkaisut ovat usein todella kattavia ja sisältävät paljon eri ominaisuuksia. Kaikki ominaisuudet eivät ole kuitenkaan välttämättä tarpeellisia, joten pienempi työkaluohjelmisto voi siis olla yrityksen kannalta joustavampi ja käytännöllisempi vaihtoehto. Eräänä vaihtoehtona on myös ilmaisen, avoimen lähdekoodin ohjelmien käyttäminen. Käyttöönotto vaihe vaatii onnistuakseen kuitenkin yrityksen sisäistä osaamista tai rahallisia panostuksia ulkopuolisen ammattitaitoisen työvoiman palkkaamiseen. Järjestelmän valinta on siis täysin tapaus- ja yrityskohtaista. Järjestelmän valintaa suorittaessaan yrityksen täytyy tarkastella omaa toimintaansa ja tarpeitaan. Ulkopuolisen konsultin palkkaamisesta voi olla apua oikean järjestelmän valinnassa tai jo hankitun järjestelmän käyttöönotossa. (Oksanen 2010, 183-188.)

2.2 Tietojen käsittely

Henkilötietojen käsittely on monivaiheinen tapahtumasarja, joka käsittää kaiken henkilötietoihin liittyvän toiminnan. Jo keräämisen suunnittelu on henkilötietojen käsittelyä. Käsitteilyyn kuuluu myös tietojen tallentaminen eri muodoissaan, tietojen muokkaaminen, kerättyjen tietojen hyödyntäminen sekä poistaminen. Tietoja ei myöskään saa käsitellä ilman asiallisia ja lainmukaisia perusteita. (Tietosuojavaltuutetun toimisto 2019.) Paperisessakin asiakastietojärjestelmässä on jo täytynyt ottaa huomioon lait ja henkilötietojen asianmukainen käsittely. Järjestelmän sähköistämisen vuoksi siihen on kuitenkin kiinnitettävä erityistä huomiota.

Sähköisessä muodossa olevien tietojen käsittelyssä tulee noudattaa erityistä huolellisuutta. EU:n lainsäädännön mukaisesti sähköisiä asiakastietoja ei saa siirtää EU:n ulkopuolelle maihin, joissa tietosuojan taso ei ole riittävä. Euroopan komissio määrittelee tietoturva-vaatimukset täyttävät maat. Asiakastietojärjestelmän hankintavaiheessa olisi hyvä kiinnittää huomiota tietokantapalvelimien fyysiseen sijaintiin, jotta tiedot eivät huolimattomuuden vuoksi päädy EU-maiden ulkopuolelle ilman asianmukaista syytä tai tarvetta. Hankintavaiheen tutkimustyöllä voidaan välttää monet ongelmatilanteet, kuten lain rikkominen ja asiakkaiden henkilötietojen vaarantuminen. (Oksanen 2010, 133-135.)

2.3 Tietoturva

Yksi sähköisen järjestelmän peruspilareista on tietoturva, jonka toimivuudesta yritys on vastuussa. Palveluna hankitun asiakastietojärjestelmän kohdalla täytyy kiinnittää erityistä huomiota palveluntarjoajan toimintaan ja tarjottavaan palveluun, sillä rekisterinpitäjä on aina jossain määrin vastuussa keräämistään sähköisistä tiedoista. (Oksanen 2010, 140.)

Tietojen sijainti omalla palvelimella sisältää myös tiettyjä haasteita turvallisuuden suhteen. Yrityksen täytyy kiinnittää huomiota palvelimen sijaintiin ja estää pääsy palvelintilaan asi-aankuulumattomilta henkilöiltä. Myöskään eläimet eivät kuulu palvelintilaan. Vaikka eläimet eivät ole sinällään tietoturvariski, niin ne voivat vaurioittaa itse palvelinta, jolle tiedot on tallennettu. Palvelimen tulisi siis sijaita tarkoitukseen sopivassa, lukitussa tilassa. (Kuutti 2011, 270-271.)

Fyysisen palvelimen turvallisuuden huolehtimisen lisäksi käytössä olevan käyttöjärjestelmänkin täytyy olla turvallinen. Tietoturva on käsitteenä laaja, mutta Kuutti listaa kuitenkin teoksessaan tiettyjä palvelimen tietoturvan perusteita. (Kuutti 2011, 271.) Seuraavaan kolme kohtaan on koostettu Kuutin listasta sellaisia perusteita, joita lähes jokaisen on mahdollista noudattaa ilman mittavaa tietoteknistä taustaa.

1. Järjestelmän päivitys ja ajantasaisuus

Verkkopalvelimeen tulee asentaa ajantasaisimmat päivitykset. Ohjelmista voi löytyä ohjelmointivirheitä, ja ne voivat potentiaalisesti vaarantaa tietoturvan. Verkko-yhteyden päässä kannattaa myös pitää vain täysin välttämätön määrä palveluita. Kaikki ylimääräiset palvelut ovat mahdollinen tietoturvariski ja lisäävät päivittämisen tarvetta. (Kuutti 2011, 271-273.)

2. Salasanakäytännöt

Hyvillä salasanoilla voidaan estää asiattomilta pääsy järjestelmiin. Hyvän salasanan käytännön mukaisesti pidemmät salasanat ovat turvallisempia kuin lyhyet. Salasanan tulee myös sisältää monipuolisesti eri merkistöä, eikä se saa olla helposti arvattavissa. Salasanoissa on myös syytä välttää selkokieliä sanoja, sillä silloin ne ovat haavoittuvaisempia automatisoiduille hyökkäyksille. (Kuutti 2011, 272.)

3. Varmuuskopiointi

Varmuuskopiointi on tärkeä osa sähköisiä järjestelmiä. Tietoa voi hävitä järjestelmästä monesta eri syystä, mutta varmuuskopiointilla voidaan välttää useat ongelmatilanteiden seuraukset. Myös varmuuskopioiden säilytyksessä tulee noudattaa samanlaista varovaisuutta kuin tietokantapalvelimen kohdalla. Varmuuskopioita on

suositeltavaa säilyttää palvelimesta erillään turvallisessa paikassa. Tällöin on epätodennäköistä, että molemmat tuhoutuvat yhtä aikaa. Myös salaamisella on mahdollista parantaa varmuuskopioien tietoturvaa. (Kuutti 2011, 274-275.)

3 NUOHOUSKORTISTO-OHJELMAN MÄÄRITTELY

3.1 Nykytilanteen kartoittaminen

Opinnäytetyö toteutettiin projektinomaisesti. Määrittely alkoi toimeksiantajan haastattelulla, jossa kartoitettiin nuohouskortiston nykytila. Myös sähköisen kortiston vaatimukset ja käyttötarkoitukset selvitettiin.

Toimeksiantajalla on käytössään täysin paperimuotoinen nuohouskortisto. Paperiset kortit täytetään käsin, ja niitä säilytetään osoitteiden mukaan aakkostettuna kannellisissa arkistolaaikoissa, lukitussa arkistokaapissa.

Yksi nuohouskortti sisältää asiakkaan nimen sekä nuohotun kiinteistön seuraavat sijaintitiedot: kaupunginosa, lähiosoite, postinumero ja postitoimipaikka. Lisäksi korttiin on merkitty kiinteistön tulisijat ja nuohouspäivämäärät. Kortin kääntöpuolelle on mahdollisuus tarvittaessa kirjoittaa nuohoustyön kannalta olennaista tietoa kiinteistöstä. Tällaista tietoa voi olla esimerkiksi merkintä tikkaiden kunnosta.

Tarve nuohouskortiston sähköistämiseksi on suuri. Toimeksiantajalta saadun tiedon mukaan käsin täytettävän kortiston ylläpitäminen on aikaa vievää ja työlästä. Yrityksen toiminnan alkuvaiheessa käsin täytettävä kortisto oli suhteellisen kätevä, sillä se ei ollut vielä kovin laaja. Asiakasmäärän kasvu on kuitenkin hankaloittanut kortiston käyttöä, sillä korttien suuri määrä vaikeuttaa tiedon hakemista, lajittelua ja arkistointia. Vähintäänkin haastavaa on esimerkiksi nuohottujen vapaa-ajan asuntojen etsiminen ja lajittelu osoitteen perusteella aakkostetusta käsikortistosta. Laajan kortiston säilytys vie paljon tilaa, ja yhden tietyn kortin löytäminenkin voi olla vaikeaa lajittelusta huolimatta. Nuohouskortti voi myös vahingossa päätyä väärään paikkaan. Pelkästään paperisessa muodossa säilytettävä kortisto on myös alttiimpi erilaisille vaaratekijöille, sillä sen varmuuskopioiminen ei ole realistisesti toteutettavissa. Juuri tämänkaltaisiin ongelmiin toimeksiantaja pyrkii löytämään ratkaisun nuohouskortiston sähköistämiseksi.

Yrityksellä on valmiiksi hankittuna yrityskäytössä ollut pöytäkone sekä sen oheislaitteet, joita voidaan hyödyntää kortiston sähköistämiprojektissa. Keskusyksikön voi muuttaa palvelinkäyttöön soveltuvaksi.

3.2 Toteutusvaihtoehdon valinta

Luvussa 2.1 käytiin lävitse pienyritykselle mahdollisesti sopivia ratkaisuja yrityksen tarpeet täyttävän tietojärjestelmäratkaisun valinnassa. Toimeksiantajan kanssa käytyjen keskustelujen ja valmiilta nuohouskortistolta odotettavien toiminnallisuuden perusteella selvisi,

että edellä mainitussa luvussa tarkastellut vaihtoehdot valmiiden ja puolivalmiiden ratkaisujen hyödyntämisestä eivät olleet yritykselle sopivia. Toteutustavaksi valittiinkin ohjelmiston tuottaminen alusta loppuun itse. Ohjelma toteutetaan verkkoselainpohjaisena ratkaisuna, jota käytetään yrityksen omalla Linux-pohjaisella palvelimella. Tietokantajärjestelmänä toimii puolestaan MariaDB.

Kyseiseen ratkaisuun päädyttiin huomioiden yrityksen toiveet, budjetti ja toteutuksen tekijän resurssit. Valmiin kuukasimaksullisen palvelun ostaminen ei olisi kustannustehokas ratkaisu, koska yrityksellä on jo valmiina palvelinkäyttöön soveltuva tietokone. Lisäksi toimeksiantaja haluaa olla täysin tietoinen siitä, missä asiakastiedot sijaitsevat fyysisesti. Tietojen sijaitseminen online-palvelimella ei tässä tapauksessa ole tarpeellista, koska se mahdollistaisi vain ylimääräisen ja turhan haavoittuvuuden tietoturvan näkökulmasta.

Valmiin ohjelmaratkaisun räätälöiminen käyttötarkoitukseen soveltuvaksi olisi myös mahdollista, mutta valmishjelmissä on nuohouskortiston kannalta paljon turhia ominaisuuksia ja tietokenttiä. Ohjelmaa täytyisi siis muokata merkittävästi, jotta siitä saataisiin toimeksiantajalle mieluinen. Yrittäjän tietotekninen tausta ei ole myöskään kovin kattava, joten vieraaseen käyttöliittymään ja ohjelman ominaisuuksiin perehdyttäminen veisi aikaa.

Edellä mainitut seikat huomioiden päädyttiin verkkoselainpohjaisen sovelluksen kehittämiseen. Sovelluksesta voidaan tehdä juuri asiakkaan tarpeet täyttävä, ja sen käytön opettelu ei myöskään tuota merkittäviä vaikeuksia, sillä sovellusta voi hallita toimeksiantajalle ennestään tutuilla verkkoselaimilla. Valituista palvelin- ja tietokantaratkaisuista on kerrottu lisää opinnäytetyön kappaleessa 5: Sähköisen nuohouskortiston toteutus.

3.3 Vaatimukset

Jotta ohjelmistoprojekti onnistuisi, sen täytyy täyttää tietyt asiakasvaatimukset. Vaatimukset jaetaan tyyppillisesti kolmeen eri ryhmään: toiminnallisiin ja ei-toiminnallisiin vaatimuksiin sekä reunaehtoihin. Toiminnallisissa vaatimuksissa kuvataan ohjelman toimintoja, kun taas ei-toiminnalliset vaatimukset keskittyvät käsittelemään esimerkiksi ohjelman ulkoasua. Reunaehdoiksi määritellään toteutuksen puitteet, kuten millä laitteilla ohjelman tulee toimia ja millä tavoilla se on toteutettava. (Haikala & Mikkonen 2011, 61-62.) Sähköisen nuohouskortistoprojektin vaatimukset kartoitetaan toimeksiantajan haastattelussa, ja niistä toteuttamiskelpoiset kirjataan ylös taulukkoon edellä mainittua ryhmittelyä noudattaen. (Taulukko 1.)

Taulukko 1. Vaatimukset

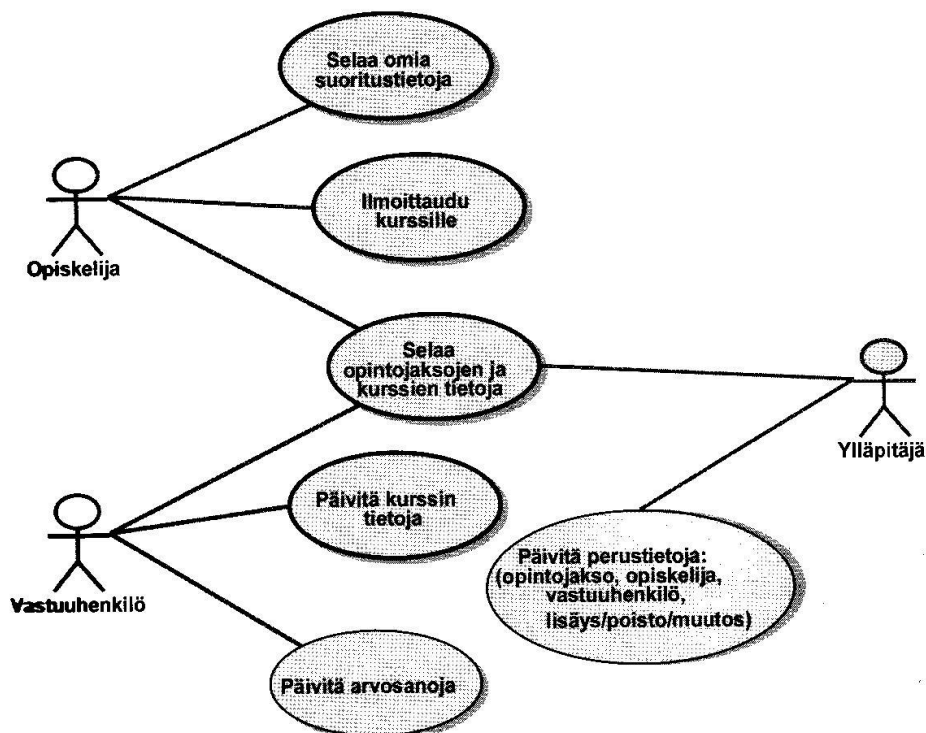
Toiminnalliset vaatimukset
Järjestelmään voi lisätä nuohouskortin
Tallennettuja nuohouskortteja voi tarkastella
Tallennettuja nuohouskortteja voi muokata
Tallennetun nuohouskortin voi poistaa
Nuohouskortteja voi hakea

Ei-toiminnalliset vaatimukset
Selkeä ja helppokäyttöinen ulkoasu
Ohjelman päävärit: musta, harmaa ja valkoinen
Toimii useilla verkkoselaimilla
Jonkinlainen tulostusmahdollisuus

Reunaehdot
Toimivuus Linux-palvelinympäristössä
Ohjelmaa voi käyttää Firefox-selaimella
Ohjelman tulee toimia toimeksiantajan laitteistoilla
Toteutettava ilmaisilla avoimen lähdekoodin/lisenssipapilla ohjelmilla

3.4 Käyttötapaukset

Käyttötapauksista piirrettiin käyttötapauskaavio (Liite 1.) Haikalan ja Mikkosen (2011) kuva soveltaen. Kaavioon piirretään tikku-ukkoina järjestelmän käyttäjät ja kupliin käyttötapaukset. (Kuvio 1.) Käyttötapauskaavion lisäksi ne kuvataan myös tekstimuodossa (Liite 2.) Haikalan ja Mikkosen (2011) esimerkin pohjalta, jossa nimetään käyttötapaus ja käyttäjät. Tarkempi kuvaus sisältää tapahtumien oletetun kulun vaihe vaiheelta sekä siihen liittyvät mahdolliset poikkeustilanteet. (Kuvio 2.) Käyttötapauksien yksinkertaisuus on yksi syy niiden toimivuudelle. Hyvä käyttötapaus on selkokielineen ja kuvaa asiat niiden tapahtumisjärjestyksessä, eikä se ole liian pitkä. Kun asia on selkeästi ymmärrettävissä, lopputuotteesta saadaan paremmin asiakkaan toiveiden ja tarpeiden mukainen. Hyvin laaditut käyttötapaukset palvelevat myös kehittäjää, koska selkeistä kuvailuista on helppo hahmotella ohjelmaan tulevia toteutusratkaisuita. Käyttötapauksien kuvauksessa ei kuitenkaan kannata olla liian yksityiskohtainen, etteivät ne turhaan rajoita ohjelman toteutusmahdollisuuksia. (Haikala & Mikkonen 2011, 79-83.) Nuohouskortistoprojektin kannalta keskeiset käyttötapaukset johdettiin toimeksiantajan haastattelun pohjalta kerätyistä toiminnallisista vaatimuksista. Liitteiden tarkat käyttötapaukset kuvaavat siis kaikki sovelluksen päätoiminnot: uuden nuohouskortin lisäämisen sekä lisättyjen korttien tarkastelun, muokkaamisen, poistamisen ja hakemisen.



Kuvio 1. Käyttötapauskaavio (Mikkonen & Haikala 2011, 77)

Nimi: Ilmoittaudu kurssille.

Versiohistoria: versio 1.0/ijh.

Osallistujat: Opiskelija

Tuloehdot: Opiskelija, kurssi ja opintojakso on syötetty järjestelmään, kurssille ilmoittautuminen on avattu, opiskelija on kirjautuneena järjestelmään (KT:t Päivitä perustietoja, Päivitä kurssin tietoja)

Kuvaus: Opiskelija seuraa WWW-linkkiä, joka johtaa kurssin sivulle. Hän valitsee vaihtoehdon ilmoittaudu. Järjestelmä tarkastaa, että opiskelijalla on tarvittavat esitiedot [Poikkeus: esitietovaatimukset eivät täyty]. Järjestelmä varmistaa vielä OK/CANCEL-kyselyllä, että opiskelija todella haluaa ilmoittautua. Tämän jälkeen järjestelmä lisää opiskelijan kurssin osallistujaksi.

Poikkeukset: Esitietovaatimukset eivät täyty: opiskelijalle annetaan luettelo puuttuvista esitietokursseista.

Lopputulokset: Opiskelija on rekisteröity kurssin osallistujaksi.

Muut vaatimukset: Päivittäin käsitellään kiireisimpänäkin tuntina enintään noin 50 varausta. Vastausajan on aina oltava alle 5 sekuntia.

Kuva 5.3: Yksi mahdollisuus käyttötapausten kuvaamiseen.

Kuvio 2. Käyttötapausten kuvaaminen (Mikkonen & Haikala 2011, 80)

4 SÄHKÖISEN NUOHOUSKORTISTON SUUNNITTELU

4.1 Sovelluksen ulkoasu

Verkkoselainpohjaisen sovelluksen suunnittelussa noudatettiin teemahaastattelussa ilmenneitä ja yhdessä toimeksiantajan kanssa sovittuja vaatimuksia. Toimivuus ja asiassällöt ovat siis kaikkein keskeisimmässä roolissa suunnittelu- ja toteutusvaiheissa. Sovelluksen ulkoasua lähdettiin suunnittelemaan yrittäjän kanssa hahmottelemalla perinteisesti paperille, miltä sovellus voi näyttää. Paperille piirrettyjen kuvien avulla toimeksiantajan näkemys oleellisista ominaisuuksista ja niiden sijainnista saatiin jo varhaisessa vaiheessa selville. Kyseiset varhaiset käyttöliittymäluonnokset antoivat hyvän pohjan itse työn toteutamiselle.

Toimeksiannon pohjalta toteutettavan nuohouskortiston päätavoitteita ovat helppokäyttöisyys ja selkeys. Käytettävyys on olennaisessa osassa, kun tuotetaan selaimella avautuvia sisältöjä. Suunnitteluvaiheessa onkin tärkeä huomioida, kuinka käytettävissä oleva näyttöala hyödynnetään, jotta sivuilla olisi mahdollisimman vähän toissijaista sisältöä. Tyhjän tilan oikeaoppinen hyödyntäminen selkeyttää sivuja ja lisää niiden helppokäyttöisyyttä sekä luettavuutta. (Nielsen 2000, 18.) Sivujen suunnittelussa ei ole myöskään suotavaa määrittellä sivujen kokoa kiinteäksi, jotta käyttäminen erikokoisilla resoluutioilla on mahdollista. Kiinteät määrittelyt voivat myös vaikeuttaa tulostusta. (Nielsen 2000, 29-30.) Resoluutio täytyy siis huomioida suunnittelussa, vaikka toimeksiantajalle toteutettavaa nuohouskortistoa ei käytetä kuin yhdeltä päätelaitteelta. Toimeksiantajan käyttämä näyttö voi vaihtua tulleisuudessa, joten sivuja ei ole järkevää tehdä toimivaksi vain yhdellä laitekoonpanolla. Lisäksi kortistosta on toisinaan tarpeellista tulostaa tietoa.

4.2 Palvelinympäristön suunnittelu

Lisenssivapaan Linux-jakelun eräs merkittävä etu on sen ilmaisuus (Kuutti 2011, 25). Linux toimii hyvin niin suurilla kuin pienilläkin alustoilla. Tämän vuoksi useat eri jakelut ovat erityisen suosittuja palvelinkäytössä. (Kuutti 2011, 23.)

Toimeksiantajan päätelaitteelle, projektia varten varattuun pöytäkoneeseen, asennetaan palvelinohjelmisto. Vaatimuksien ja määrittelyn perusteella päädyttiin ilmaisen Linux-jakelun käyttämiseen. Toimeksiantajalla ei ole myöskään suunnitelmassa käyttää sovellusta etänä, joten palvelinta ei ole tarkoituksenmukaista kytkeä verkkoon.

4.3 Tietokanta ja siihen siirrettävät tiedot

Sähköisen nuohouskortiston tietokannan sisällöt noudattelevat asiakkaan alkuperäisen paperikortiston tietoja. Tietokantaan lisätään siis vain nuohoustyön suorittamisen kannalta oleellisia tietoja. Vanhasta kortistokortista ja yrittäjän kanssa tehdyistä suunnitelmista ilmenevät seuraavat tietokantaan tallennettavat tiedot:

- asiakkaan etunimi
- asiakkaan sukunimi
- kaupunginosa
- lähiosoite
- postinumero
- postitoimipaikka
- tulisijat
- nuohousajankohdat
- asunnon käyttötarkoitus
- lisätietoja.

Tietokenttiä varten nuohouskortiston tietokantaan luodaan Asiakkaat-taulu, johon tiedot kerätään omiin kenttiinsä. Suurin osa kentistä määritellään niin, etteivät ne hyväksy tyhjiä arvoja. Ylimääräisiä tulisijoja ja niiden nuohouspäivämääriä sekä Lisätietoja-kenttää ei ole pakko täyttää, vaan ne voidaan jättää tyhjiksi. Kaikki tarvittavat tiedot voidaan tallentaa yhteen tauluun, joten tietokannan rakenne pysyy yksinkertaisena.

1.1.2019 voimaan tullut laki pelastuslain muuttamisesta ei enää velvoita nuohoojaa säilyttämään nuohouskortteja kymmenen vuoden ajanjaksoa (Pelastuslaki 379/2011, 62 §; Laki pelastuslain muuttamisesta 1078/2018). Uuteen sähköiseen tietokantaan on lakimuutoksen vuoksi luontevaa siirtää vain vuoden 2019 alusta kertyneet asiakastiedot. Vanhan lain säilytysajan mukainen nuohouskortisto pidetään erillään, ja se arkistoidaan lain velvoittamaksi ajaksi erilliseen lukittuun tilaan. Vanhassa kortistossa on mahdollisesti myös passiivisia asiakkuuksia, ja osa korteista voi olla jo lähes kymmenen vuoden takaa. Kortistonpidon selkeyttämiseksi vanhan lain mukainen paperinen kortisto on siis hyvä pitää erillään. Aktiiviset asiakkuudet tulevat kuitenkin siirtymään sähköiseen kortistoon 1-3 vuoden sisällä järjestelmän käyttöönotosta.

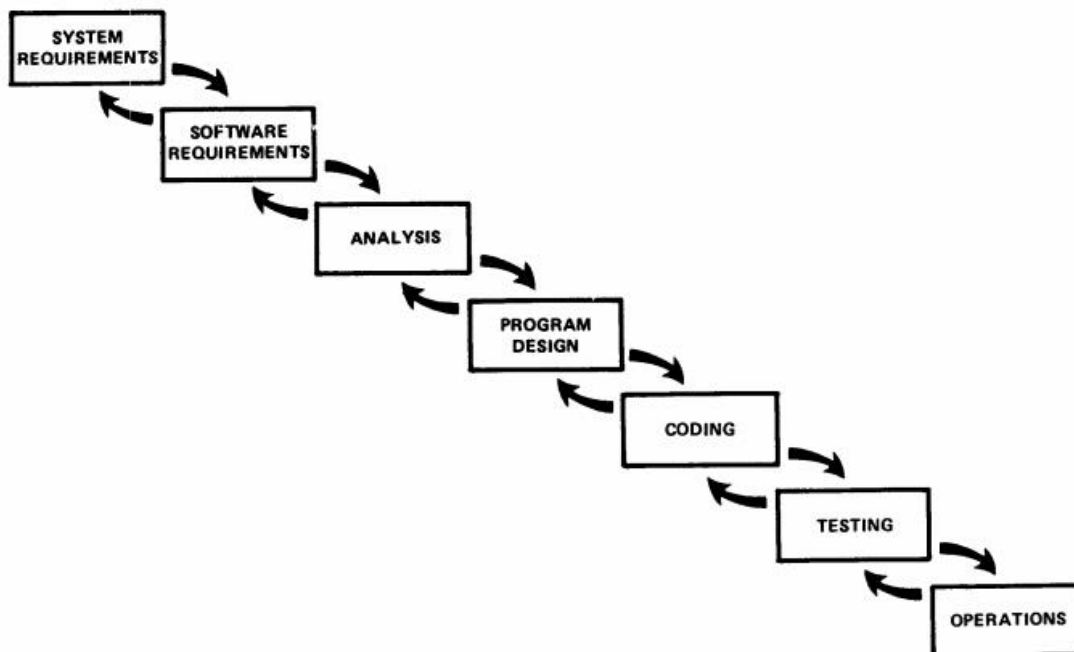
5 SÄHKÖISEN NUOHOUSKORTISTON TOTEUTUS

5.1 Projektinhallintamalli

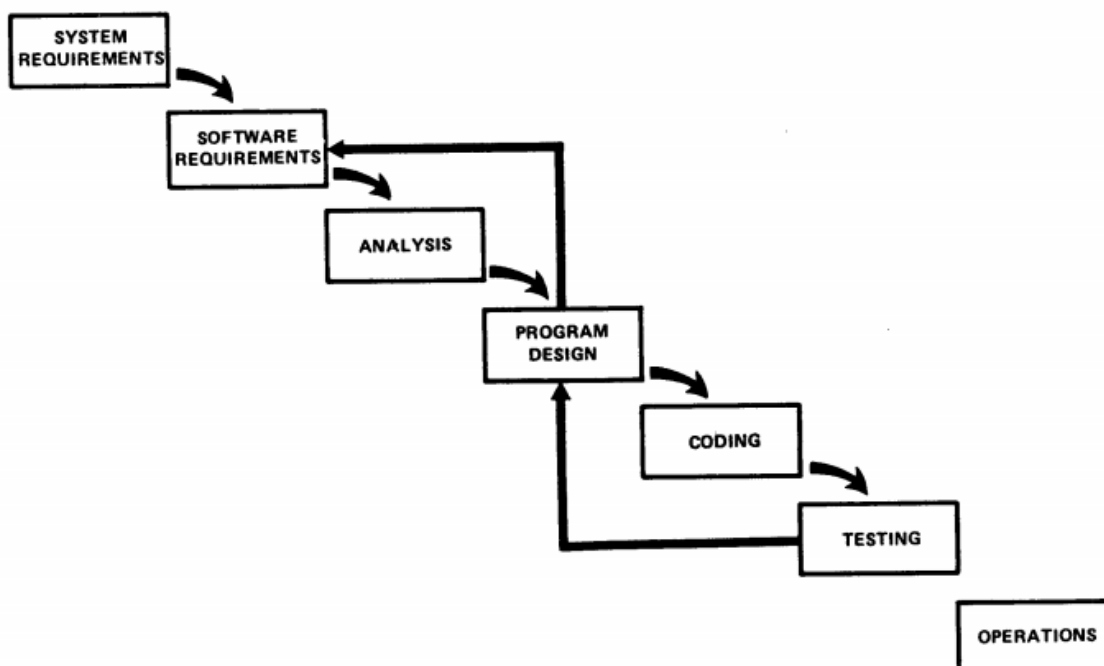
Projektin toteutus eteni Roycen muokatun vesiputousmallin mukaisesti. Roycen vesiputousmalli kuvaa projektin kaikki vaiheet järjestelmän vaatimuksien määrittelystä aina käyttöönottoon. Eri vaiheet on aseteltu vesiputousta muistuttavaan muotoon, ja nuolet kuvaavat seuraavaan vaiheeseen siirtymistä. Mallissa on kuitenkin mahdollisuus palata aiempaan vaiheeseen tarpeen vaatiessa, mitä toiseen suuntaan piirretyt nuolet kuvaavat. (Kuvio 3.) Alkuperäinen vesiputousmalli ei sisällä aikaisempiin vaiheisiin palaamista, joten ongelmat voivat ilmetä vasta testausvaiheessa. Ongelmien ilmeneminen vasta projektin loppuvaiheessa voi vaatia jopa koko sovelluksen uudelleensuunnittelua ja -toteutusta, jolloin projektin alkuperäinen aikataulu ja kustannusarvio eivät toteudu. (Kuvio 4.) Roycen alkuperäinen malli mahdollistaakin projektin mahdollisten ongelmien havaitsemisen varhaisemmassa vaiheessa. (Royce 1970.)

Projektin olisi voinut toteuttaa myös jollain puhtaasti iteratiivisella mallilla. Iteratiivisessa toimintamallissa useiden eri ohjelmaversioiden eli iteraatioiden kehittäminen ja asiakkaalta kerätty palaute eri versioista on erittäin keskeisessä roolissa. Runsaalla iteroinnilla on hyötyjen lisäksi myös riskejä. Runsas testaaminen ja riskienhallinnan monimutkaistuminen voi pidentää projektin kestoja. (Haikala & Mikkonen 2011, 40-42.)

Nuohouskortiston toteuttamisessa useiden eri versioiden tuottaminen ja toimeksiantajalla kokeiluttaminen eivät olisi olleet realistisia vaihtoehtoja huomioiden projektin kesto, laajuus ja käytettävissä olevat henkilötyötunnit. Toimeksiantajalla olisi myös pitänyt olla aikaa perehtyä huolellisesti useisiin versioihin muiden töidensä ohessa sekä antaa niistä tarpeeksi rakentavaa palautetta, jotta ohjelmaa olisi voitu jatkokehittää. Roycen vesiputousmalli soveltui siis tähän projektiin hyvin täysin iteratiivisen mallin sijaan. Aikaisempiin vaiheisiin palaamisella voidaan välttää alkuperäisen vesiputouksen suurimmat riskit. Toimeksiantajaan oltiin tiiviisti yhteydessä eri vaiheiden aikana, ja tarvittaessa palattiin aiempaan vaiheeseen tekemään muutoksia toivomusten pohjalta.



Kuvio 3. Roycen (1970) muokattu vesiputousmalli.



Kuvio 4. Roycen (1970) kuvaus perinteisen vesiputousmallin ongelmasta, jossa ongelmat voivat ilmetä vasta projektin loppuvaiheessa.

5.2 Laitteet ja ohjelmistot

Sähköisen nuohouskortiston toteuttamisessa hyödynnettiin ja käytettiin useita eri laitteita ja ohjelmistoja. Seuraavaksi käyn niistä lävitse projektin toteutuksen kannalta kaikkein oleellisimmat, ja lisäksi perustelen suoritettuja valintoja.

Tietokoneet

Toimeksiantajalla oli projektia varten käytössään pöytäkone. Pöytäkoneen valintaan päädyttiin kannettavan tietokoneen sijaan toimintavarmuuden ja oheislaitteiden, kuten näytön helpomman vaihtamisen vuoksi. Toimeksiantajan mielestä muun muassa suurempi näyttökoko tekee pöytäkoneen käytöstä miellyttävämpää. Kone on myös riittävän tehokas, jotta sitä voitiin hyödyntää palvelinkäytössä.

Ohjelmakoodin kirjoitin suurimmaksi osaksi omalla kannettavalla tietokoneellani. Ohjelman testaamista varten loin toimeksiantajan pöytäkonetta vastaavan virtuaalikoneen, jotta ohjelman toimintaa ei ollut aina pakollista testata paikan päällä. Työtä tehtiin kuitenkin myös valmiilla palvelimella, jotta ohjelman toimivuudesta voitiin varmistua.

Notepad++

Notepad++ on ilmainen avoimen lähdekoodin koodieditori. Se sisältää monia tuotantoa helpottavia ominaisuuksia, kuten virheiden korostamisen. (Don Ho 2019.) Olen todennut Notepad++ -ohjelman hyödylliseksi aiemmissa projekteissani, joten sen käyttäminen tässäkin tapauksessa tuntui luontevalta ja työskentelyä tehostavalta.

CentOS 7

CentOS on ilmainen Linux-käyttöjärjestelmän jakelu, ja se perustuu palvelinkäytössä yleiseen Red Hat Linuxiin. Suositulla jakelulla on vahvat näytöt toimivuuden ja vakauden osalta verrattuna moniin kilpailijoihin. (Kuutti 2011, 16.) Sähköisen nuohouskortiston toteuttamiseen käytettiin CentOS 7 -Linux-jakelua juuri sen palvelinkäytön kannalta hyvien ominaisuuksien vuoksi. Kyseisen käyttöliittymän graafinen näkymä on myös helppokäyttöinen ja selkeä, joten se on toimeksiantajankin kannalta hyvä valinta.

MariaDB

MariaDB on MySQL:n entisten kehittäjien kehittämä tietokantasovellus. Se on nopea avoimen lähdekoodin tietokantaratkaisu, jossa on runsaasti ominaisuuksia. (MariaDB Foundation 2019.) Valitsin MariaDB:n, koska tietokannan pysyminen ilmaisena on yritykselle tärkeää. Käytettävyydeltään se on kuitenkin hyvin samankaltainen kuin MySQL, joten sen käyttäminen ei vaikeuttanut työn toteuttamista.

Apache

Apache on avoimen lähdekoodin web-palvelinohjelmisto, jonka avulla voidaan pyörittää web-sivuja ja sovelluksia (Apache 2019). Apachen valintaan päädyin aikaisemman käyttökokemuksen perusteella.

EPEL

EPEL eli Extra Packages for Enterprise Linux on Fedora-ryhmän keräämä ja ylläpitämä kokoelma erilaisia Linuxin yrityskäyttöön soveltuvia hyödyllisiä sovelluksia. EPEL-paketin asentamisen jälkeen haluamansa sovellukset voi asentaa omaan Linux-jakeluunsa. (Fedora Project Wiki 2019.) EPEL-paketti on projektin kannalta tärkeä, koska se sisältää phpMyAdminin, jota ei löydy CentOS 7-jakelusta oletuksena.

phpMyAdmin

phpMyAdmin on ilmainen työkalu, jolla voidaan hallita tietokantaa verkkoselaimen avulla. Se tukee useita eri MySQL:n ja MariaDB:n toimintoja, jotka ovat työskentelyn helpottamisen ja tehostamisen kannalta hyödyllisiä. (phpMyAdmin 2019.) Valitsin phpMyAdminin aiemman käyttökokemuksen, tarpeellisten ominaisuuksien sekä helppokäyttöisen selainnäköisen vuoksi.

HTML5

HTML eli hypertext markup language on merkintäkieli, jonka avulla voidaan kuvata verkkosivujen rakenne. Verkkoselain käyttää HTML:ää verkkosivun muodostamisessa. (w3schools 2019a.) Tällä hetkellä uusimmassa HTML5-standardissa ei käytetä kaikkia samoja elementtejä kuin vanhassa HTML4-standardissa. Uudessa versiossa on myös useita uusia elementtejä, esimerkiksi ylä- ja alatunnisteita sekä multimediaa varten. (w3schools 2019b.) HTML:än käyttö oli luonteva ratkaisu, koska nuohouskortisto toteutetaan verkkoselaimella toimivana sovelluksena. Laadin tiedostot uusimman HTML5-standardin mukaisesti, jotta ne toimivat mahdollisimman hyvin uusimpien verkkoselaimien kanssa.

CSS

CSS eli Cascading Style Sheets on tyylitiedosto, joka kuvaa kuinka HTML-merkintäkielen elementit näkyvät ruudulla. Tyylitiedoston käytöllä voidaan vähentää turhaa työtä, koska yhdellä tiedostolla voidaan määritellä koko verkkosivun asettelu ja muotoilu. (w3schools 2019c.) Valitsin CSS-tyylitiedoston käytön, koska se toimii luontevasti HTML-merkkikielen parina. Tyylitiedoston käyttäminen säästää aikaa, ja sen avulla nuohouskortin ulkoasusta saadaan paljon miellyttävämpi.

PHP

PHP eli Hypertext Preprocessor on erityisesti web-kehitykseen soveltuva monikäyttöinen ohjelmointikieli. PHP-kielellä voidaan suorittaa toimintoja suoraan palvelimessa, jolloin loppukäyttäjän päähän tulee näkyviin vain koodin mukaisesti luotu HTML. Sen käyttäminen on helppoa aloittelijoillekin, mutta se sopii myös ammattilaisille. (PHP 2019.) PHP:n valintaan päädyin siis sen hyvien web-kehitysominaisuuksien ja helppokäyttöisyyden vuoksi.

5.3 Linux-pohjaisen palvelimen käyttöönotto

Pöytäkone muunnettiin palvelinkäyttöön soveltuvaksi asentamalla siihen ensin CentOS 7-jakelu. Käyttöjärjestelmä tarvitsee käyttäjiä, joten loin sille kaksi erillistä käyttäjää. Toinen käyttäjistä sai järjestelmänvalvojan oikeudet ja toinen käyttäjistä pelkät perusoikeudet. Tällä menettelyllä toimeksiantajan ei tarvitse pelätä tekevänsä mitään merkittäviä ja mahdollisesti haitallisia muutoksia järjestelmään vahingossa, silloin kun hän haluaa käyttää pelkästään sovellusta. Asetin molemmille käyttäjille myös hyvät ja vahvat salasanat. Käyttöjärjestelmän asentamisen ja käyttäjien luomisen jälkeen asensin laitteeseen seuraavat edellä mainitut ohjelmat, jotka tulevat jakelun mukana: Apache web-palvelin, PHP ja MariaDB. Toteutuksen kannalta oleellinen phpMyAdmin ei sisälly oletuksena CentOS 7 -jakelun ohjelmapakettivarastoon, joten sen käyttöönottamiseksi asensin ensin EPEL-paketin. Asentamisen ja tarvittavien tunnuksien luomisen jälkeen testasin vielä lopuksi palvelimen sekä tietokannan toimivuuden.

5.4 Tietokanta

Projektin toteutuksessa tarvittavat tietokannat tein suoraan phpMyAdminin graafisella käyttöliittymällä. Nimesin tietokannan nuohouskortistoksi, ja loin siihen Asiakkaat-nimisen taulun, johon lisäsin luvun 4.3 suunnitteluvaiheessa päätetyt kentät. Loin myös toisen taulun sovelluksen käyttäjätietoja varten. Tauluun tallentuvat tehtyjen käyttäjätilien käyttäjänimet sekä salasanat.

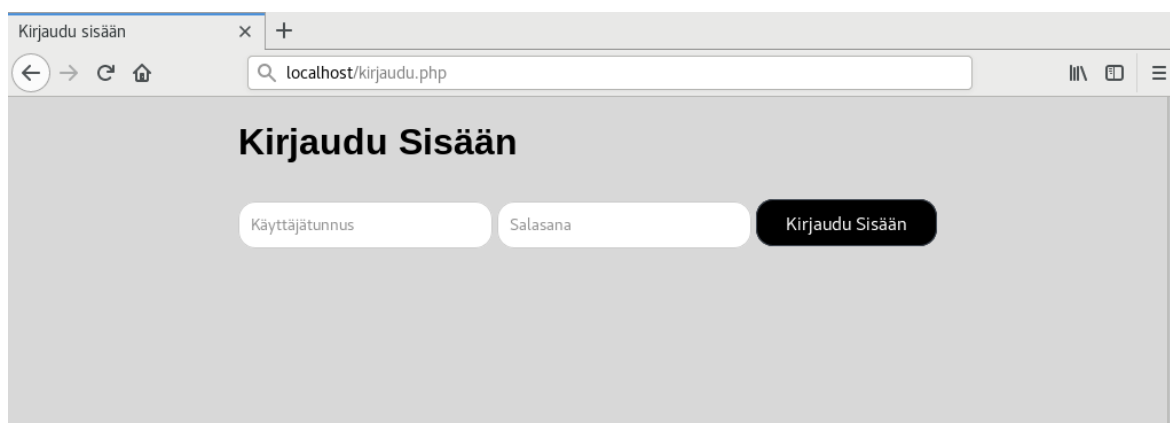
5.5 Sovellus

Hyödynsin sovelluksen toteuttamisessa luvussa 5.2 mainittuja ohjelmia ja laitteistoja. Sovelluksen kirjoittamisessa käytin HTML-merkkikieltä selainnäkömään toteuttamiseen sekä CSS-tyylitiedostoa sen ulkoasun muokkaamiseen. Luvussa 3.3 käytiin lävitse vaatimusten tärkeys projektin onnistumisessa. Pyrinkin ohjelman toiminnoilla täyttämään toiminnalliset vaatimukset. Lisäksi noudatin vaatimusmäärittelyssä päätettyjä mustan ja harmaan sävyjä

ohjelman värien valinnassa. Luvussa 4.1 puolestaan todettiin näyttöalan käytön vaikuttavan sovelluksen selkeyteen. Tein siis sovelluksesta mahdollisimman selkeän pitämällä ulkoasun tarkoituksella hillittynä. Jokaisella sivulla on selkeästi näkyvissä vain tarpeelliset tiedot. PHP-ohjelmointikieltä tarvitsin, kun tein sovellukseen ominaisuuksia, joilla tietoa voi tallentaa, poistaa, näyttää sekä hakea tietokannasta. Seuraavien alaotsikkojen alla esittelen ohjelman toiminnallisuudet.

Kirjaudu sisään

Ohjelma aukeaa työpöydän kuvakkeesta sisäänkirjautumisikkunaan. (Kuva 1.) Siihen käyttäjä syöttää käyttäjätunnuksen ja salasanan, jonka jälkeen aloitusvalikko aukeaa.



Kuva 1. Sisäänkirjautumisikkuna

Aloitusvalikko

Ohjelman päänäkyvä on selkeä, eikä se sisällä mitään turhia ominaisuuksia. Kaikki tärkeimmät valikot ovat helppokäyttöisiä, koska ne ovat selkeästi esillä. (Kuva 2.)



Kuva 2. Nuohouskortiston aloitusvalikko

Syötä uusi nuohouskortti

Sähköisen nuohouskortiston yksi tärkeimmistä ominaisuuksista on mahdollisuus lisätä uutta tietoa tietokantaan. Pyrinkin tekemään uuden kortin syöttämisestä mahdollisimman helppoa. Kaikki tarvittavat teksti- ja numerokentät on nimetty selkeästi, ja päivämäärän voi halutessaan valita kalenterista kirjoittamisen sijaan. Kun korttiin on täytetty kaikki tarvittavat tiedot, niin Tallenna-nappia painamalla PHP-koodilla toteutettu tiedon tallennusominaisuus tallentaa kaikki lomakkeen tiedot palvelimen tietokantaan. Määrittelin myös kaikki tarvittavat kentät pakollisiksi, eikä lomaketta voi tallentaa, jos tällainen kenttä on tyhjä. Edellä mainitussa tilanteessa käyttäjä saa ilmoituksen, jossa pyydetään täydentämään puuttuvat tietokentät. (Kuva 3. & Kuva 4.)

Syötä uusi nuohouskortti x +

localhost/syota.php

Aloitusvalikko Tarkastele ja muokkaa nuohouskortteja Hae Kirjaudu ulos

Syötä uusi nuohouskortti

Asiakas

Syötä etunimi Syötä sukunimi

Kiinteistön tiedot

Syötä kaupunginosa Syötä lähiosoite

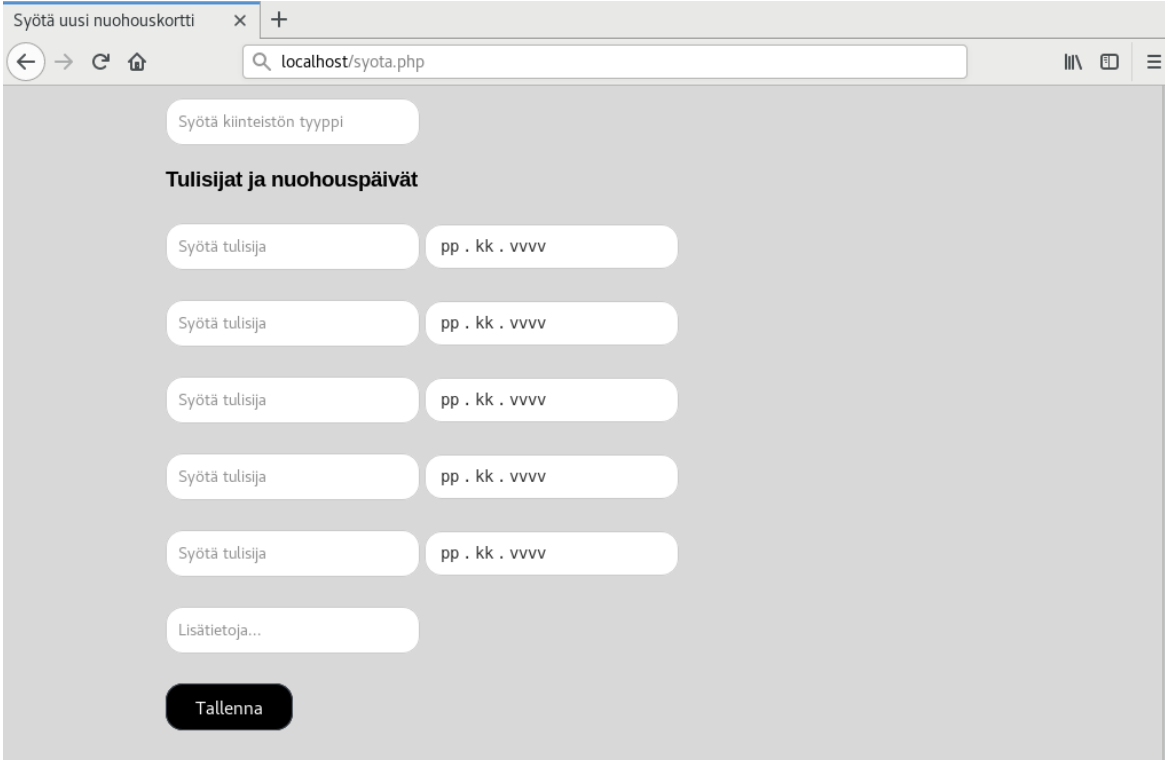
Syötä postinnumero Syötä postitoimipaikka

Syötä kiinteistön tyyppi

Tulisijat ja nuohouspäivät

Syötä tulisija pp . kk . vvvv

Kuva 3. Syötä uusi nuohouskortti -sivun yläosa



Syötä uusi nuohouskortti

localhost/syota.php

Syötä kiinteistön tyyppi

Tulisijat ja nuohouspäivät

Syötä tulisija pp . kk . vvvv

Syötä tulisija pp . kk . vvvv

Syötä tulisija pp . kk . vvvv

Syötä tulisija pp . kk . vvvv

Syötä tulisija pp . kk . vvvv

Lisätietoja...

Tallenna

Kuva 4. Syötä uusi nuohouskortti -sivun alaosa

Tarkastele ja muokkaa nuohouskortteja

Nuohouskorttien tarkastelu- ja muokkausominaisuudessa PHP-koodi noutaa tietokannasta täytetyt nuohouskortit siinä järjestyksessä, kuin ne on tallennettu järjestelmään. (Kuva 5.) Näkymässä voi siirtyä kortin muokkainäkymään sekä poistaa haluamiaan kortteja klikkaamalla taulukon oikean laidan Muokkaa tai Poista -painikkeita. (Kuva 6.) Koodi suorittaa halutun toimenpiteen tietokannassa ja avaa muokkausvalikon tai poistaa halutut tiedot. Poista-nappia painamalla ei kuitenkaan voi poistaa kortteja vahingossa, sillä poistaminen täytyy ensin vahvistaa. (Kuva 7.) Korttien muokkaus ja poistomahdollisuus itse sovelluksessa on hyödyllinen ominaisuus, jottei toimeksiantajan tarvitse suorittaa kyseisiä toimenpiteitä phpMyAdminin monimutkaisemman käyttöliittymän kautta.

Tarkastele nuohouskortteja x +

localhost/tarkastele.php

Aloitusvalikko Syötä uusi nuohouskortti Hae Kirjautu ulos

Tarkastele ja muokkaa nuohouskortteja

#	Etunimi	Sukunimi	Kaupunginosa	Lähiosoite	Postinumero	Postitoimipaikka	Tulisija 1	Päivämäärä	Tulisija 2	Päivä
1	Testi	Testaaja	Testikaupunginosa	Testikatu 13	23456	Testilä	Varaava takka	2019-03-11	Nostettava pata	2019
2	Esimerkki	Asiakas	Esimerkkilä	Esimerkkitie 8	12345	Esimerkki	Kiuas	2019-03-14	Liesi	2019
3	Pekka	Piippu	Testi	Testitie 4	23456	Testilä	Lämmitysuuni	2019-03-19	Kiuas	2019
4	Satu	Savupiippu	Esimerkkilä	Esimerkkitie 5	12345	Esimerkki	Avotakka	2019-03-20		
5	Lauri	Lämmitysuuni	Maillikaupunginosa	Mielikuvitustie 4	11223	Mallila	Kiuas	2019-03-21		

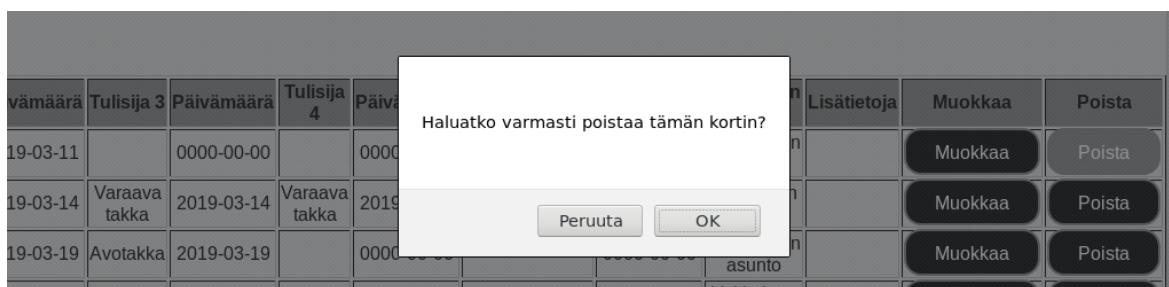
Kuva 5. Tarkastele ja muokkaa nuohouskortteja -näkyvä

Tarkastele nuohouskortteja x +

localhost/tarkastele.php

vämäärä	Tulisija 3	Päivämäärä	Tulisija 4	Päivämäärä	Tulisija 5	Päivämäärä	Kiinteistön tyyppi	Lisätietoja	Muokkaa	Poista
19-03-11		0000-00-00		0000-00-00		0000-00-00	Vapaa-ajan asunto		Muokkaa	Poista
19-03-14	Varaava takka	2019-03-14	Varaava takka	2019-03-14	Takkaleivinuuni	2018-03-14	Vakituinen asunto		Muokkaa	Poista
19-03-19	Avotakka	2019-03-19		0000-00-00		0000-00-00	Vapaa-ajan asunto		Muokkaa	Poista
							Vakituinen asunto		Muokkaa	Poista
							Vapaa-ajan asunto		Muokkaa	Poista

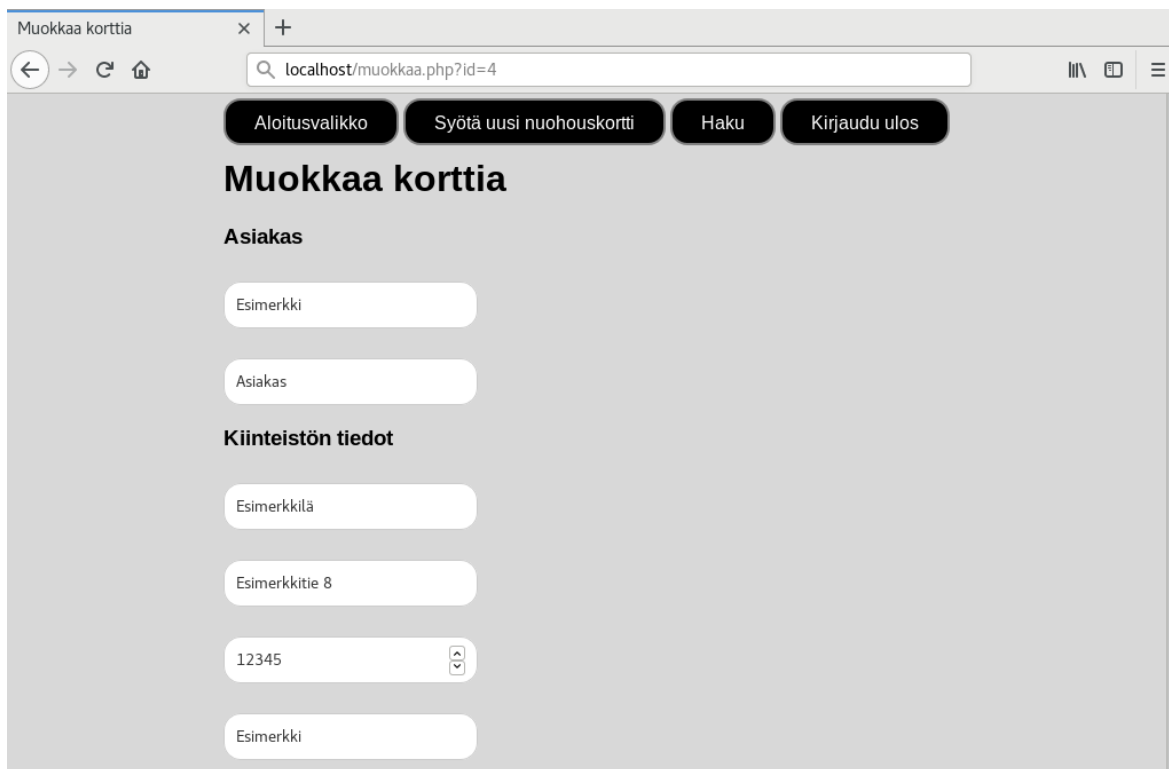
Kuva 6. Muokkaa ja Poista -painikkeet



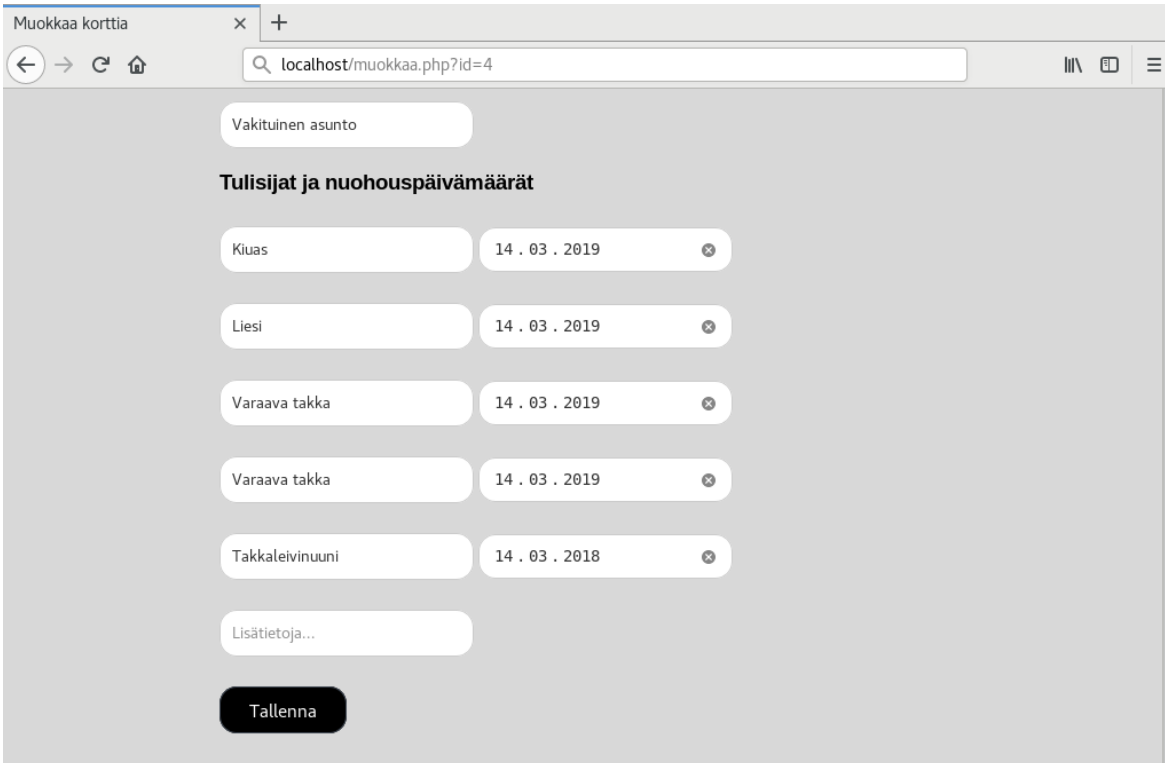
Kuva 7. Kortin poistamisen varmistusikkuna

Muokkaa korttia

Nuohouskortin muokkausvalikkoon aukeaa valitun kortin tiedot. Näkymässä on helppo päivittää haluamansa tietokentät. Tehdyt muutokset tallennetaan painamalla alalaidan Tallenna-painiketta, ja ohjelma ilmoittaa tietojen onnistuneesta tallennuksesta. (Kuva 8. & Kuva 9.)



Kuva 8. Muokkaa korttia -sivun yläosa



Muokkaa korttia

localhost/muokkaa.php?id=4

Vakituinen asunto

Tulisijat ja nuohospäivämäärät

Kiuas	14 . 03 . 2019	✕
Liesi	14 . 03 . 2019	✕
Varaava takka	14 . 03 . 2019	✕
Varaava takka	14 . 03 . 2019	✕
Takkaleivinuuni	14 . 03 . 2018	✕

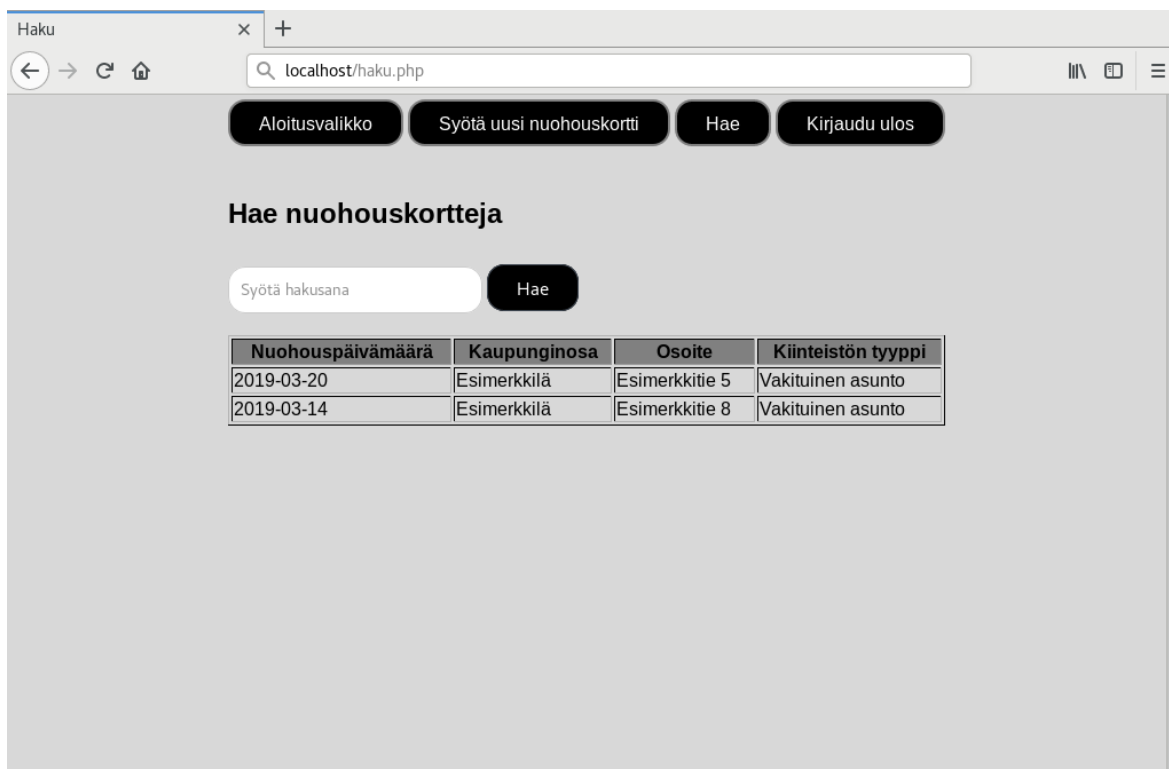
Lisätietoja...

Tallenna

Kuva 9. Muokkaa korttia -sivun alaosa

Hae nuohouskortteja

Hakutoiminnolla voi hakea hakusanalla tietoa tietokannasta kiinteistön nuohospäivän, kaupunginosan, osoitteen tai kiinteistön tyyppiin liittyvillä hakutermeillä. Hakusana syötetään hakupalkkiin, ja ohjelmakoodi noutaa tiedot tietokannasta hakupainiketta painamalla. Kyseinen ominaisuus oli toimeksiantajan toivoma, koska sillä on helppo listata tietyt kriteerit täyttävät kohteet kaikkien korttien joukosta. (Kuva 10.)



Kuva 10. Hakutoiminnolla on haettu kaikki Esimerkkitien nuohouskohteet

5.6 Testaus palvelinympäristössä

Kun sain työn valmiiksi, lisäsin sovelluksen valmiit merkki- ja kooditiedostot palvelimen /var/www/html -kansioon. Lisäksi tein työpöydälle pikakuvakkeen sovelluksen avaamisen helpottamiseksi.

Testauksessa hyödynnettiin työn aikana luotuja käyttötapauksen kuvauksia. Käyttötapauksia ja niiden poikkeustilanteita testattiin syöttämällä ja käyttämällä järjestelmässä esimerkkietietoja. Lopputestauksen aikana havaitsin muutamia kirjoitusvirheitä, mutta itse toiminnallisuudessa ei ollut ongelmia. Korjasin havaitut virheet, ja testasin vielä sovelluksen ja sen tietojen varmuuskopiointia sekä palauttamista palvelinympäristöön. Kun kaikki havaitut virheet korjattiin ja käyttötapaukset saatiin suoritettua onnistuneesti, voitiin ohjelma todeta käyttövalmiiksi.

5.7 Käyttöönotto

Sähköinen nuohouskortisto otettiin käyttöön toimeksiantajan perehdyttämisen jälkeen. Tein sovelluksesta niin helppokäyttöisen, ettei erilliselle käyttöohjeelle ollut tarvetta. Toi-

meksiantaja oli mukana koko kehitysprojektin ajan, joten hänellä oli jo valmiina hyvä käsitys siitä, kuinka sähköistä kortistoa käytetään. Esittelin yrittäjälle kaikki sähköisen kortiston toiminnot esimerkkitapauksilla, jonka jälkeen seurasin, kun hän itse harjoitteli ohjelman käyttöä. Käyttöönoton perehdytyksen yhteydessä kävimme myös läpi, kuinka kortistosta voi halutessaan tulostaa tietoa käyttämällä verkkoselaimen omaa tulostusvaihtoehtoa.

5.8 Varmuuskopiointi ja tietoturva

Kuten luvussa 2.3 todettiin, palvelimen tietoturvasta huolehtiminen on erittäin tärkeää. Nuohouskortistoprojektin kannalta oleellisia asioita luvussa ovat erityisesti varmuuskopioiden ottaminen, niiden asiallinen säilyttäminen sekä hyvien salasanojen asettaminen järjestelmiin. Sen vuoksi opastin toimeksiantajaa ottamaan varmuuskopiot ulkoiselle kovalevylle sekä palauttamaan ohjelman ja sen tietokannan varmuuskopioista. Toimeksiantajaa opastettiin ottamaan varmuuskopioita säännöllisesti noin kahden viikon - kuukauden välein. Ylimääräiseen varmuuskopiointiin kehoitettiin myös silloin, jos järjestelmään syötetään kerralla poikkeuksellisen paljon tietoa. Täydet varmuuskopiot otetaan koko käyttöjärjestelmästä, jotta palauttaminen voidaan tarvittaessa tehdä kokonaan toiselle tietokoneelle. Uuden koneen käyttöjärjestelmän asennuksen yhteydessä varmuuskopion sisältävä levy kytketään koneeseen, ja sitä käytetään asennuksessa. Käyttöjärjestelmän asennus on ohjattu prosessi, josta toimeksiantajan tulisi selviytyä annettujen ohjeiden perusteella. Opastin myös toimeksiantajaa varmuuskopioiden asiallisesta säilyttämisestä, jotta tietoturva toteutuisi mahdollisimman hyvin. Otetut varmuuskopiot säilytetään lukollisessa tilassa eri sijainnissa kuin palvelinkone. Näillä toimilla toimeksiantaja pyrkii varmistamaan sen, että varmuuskopiot pysyvät mahdollisimman hyvässä tallessa ja käyttökelpoisina.

Varmuuskopioiden ja palvelimen asiallisen säilyttämisen lisäksi tietoturvasta on huolehdittu myös muilla tavoilla. Käyttöjärjestelmän kaikki käyttäjät on salasanasuojattu aiemmin luvussa 2.3 käsiteltyjen salasanaikäytäntöjen mukaisesti. Palvelinta ei ole myöskään kytketty verkkoon, joten verkon välityksellä tapahtuvat hyökkäykset eivät ole uhka. Järjestelmän päivittämiselle ei siis ole varsinaista tarvetta, mutta esimerkiksi selainpäivityksiä voi halutessaan suorittaa ulkoisella tallennusvälineellä, kuten muistitikulla.

6 YHTEENVETO

Olen tyytyväinen työn kirjalliseen ja toiminnalliseen osuuteen. Työ vastaa tarkoitustaan, sillä sen lopputulos on toimeksiantajan tarpeet täyttävä sähköinen nuohouskortisto. Toteutus pysyi hyvin asetetuissa aikatavoitteissa ja vastasi suurimmaksi osaksi projektin suunnitelmia sekä sille asetettuja vaatimuksia. Helppokäyttöisen sähköisen kortisto-ohjelman määrittelyn, suunnittelun ja toteutuksen lisäksi selvitin eräitä tärkeimpiä seikkoja, joita pienyrityksen tulee huomioida sähköisen järjestelmän valinnassa ja käyttöönotossa.

Työskentelyn aikana esiintyneet vaikeudet liittyivät lähinnä ohjelman tekovaiheeseen. Testatessani ohjelmaa kohtasin muutamia ongelmia, joiden vuoksi kaikki sovelluksen ominaisuudet eivät toimineet toivotulla tavalla. Ongelmat johtuivat kuitenkin lähinnä kirjoitusvirheistä, joten niiden korjaaminen onnistui helposti. Muita haasteita oli puutteellinen tietämykseni nuohousalasta. Itselleni ei ollutkaan suunnitteluvaiheessa täysin selvää, mitä kaikkia tietoja sähköiseen kortisto-ohjelmaan täytyi sisällyttää. Toimeksiantajan kanssa tehty yhteistyö kuitenkin selvensi näkemystä.

Toimeksiantajan kanssa käytiin lopuksi vapaamuotoinen haastattelu, jossa tiedustelin hänen mielipiteitään työn lopputuloksesta. Toimeksiantaja on lopputuotteeseen tyytyväinen, ja ohjelma toimii hänen odottamallaan tavalla. Toimeksiantajaa miellytti erityisesti ohjelman helppokäyttöisyys, selkeä ulkoasu sekä hakuominaisuus. Nuohouskorttien täyttämisen ja hallinta verkkoselaimella käytettävällä ohjelmalla on nopeampaa verrattuna vanhaan paperiseen kortistoon. Sähköinen nuohouskortisto täytti myös sille asetetut vaatimukset. Tosin tiedon syöttämisestä ja muokkaamisesta olisi voinut tehdä vieläkin helpompaa muuttamalla Tulisija-tekstikentät pudotusvalikoiksi. Idea pudotusvalikoista syntyi vasta käyttöönottovaiheessa, joten muutoksia ei enää tehty. Toimeksiantajan käyttämä Firefox-verkkoselain osaa kuitenkin ehdottaa aiemmin syötettyjä tulisijoja, joten tekstikenttienkin täyttö on ensimmäisen täyttökerran jälkeen lähes yhtä nopeaa kuin pudotusvalikoiden käyttäminen. Valmis ohjelma on pyritty testaamaan hyvin. Mikäli ongelmia kuitenkin ilmenee, korjaan ne hyvien käytäntöjen mukaisesti kohtuullisessa ajassa.

Olisin voinut kehittää työtä pidemmälle lisäämällä sähköiseen kortistoon uusia elementtejä Javascriptillä. Silloin nuohouskorttiin voisi esimerkiksi lisätä ja poistaa tekstikenttiä reaaliaikaisesti täyttämisen aikana. Javascriptin käyttö on kuitenkin itselleni suhteellisen vierasta, eikä tarkempi perehtyminen siihen olisi tässä työssä ollut mielekäästä huomioiden työhön liittyvät aikarajoitteet.

Kirjoitushetkellä toimeksiantaja on vasta ottanut sähköisen nuohouskortiston käyttöönsä, joten jatkokehitys- ja parannusehdotusten määrä on vähäinen. Yrittäjällä heräsi kuitenkin

kiinnostus laajentaa ohjelmaa kattamaan mahdollisesti myös raportointia. Nuohoojan tarvitsee toisinaan tehdä raportteja kiinteistön tulisijojen kunnosta, ja sovellukseen lisättävä Raporttipohjat-ominaisuus voisikin noutaa kiinteistön ja asiakkaan tiedot täytetyistä nuohouskorteista valmiille raporttipohjalle. Varsinaisista jatkokehityssuunnitelmista ei kuitenkaan tässä vaiheessa ole vielä sovittu.

LÄHTEET

Painetut lähteet

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum.

Kuutti, W. 2011. Linux-käsikirja. Jyväskylä: WSOYpro Oy.

McNiff, J. & Whitehead, J. 2000. Action research in organisations. London: Routledge.

Nielsen, J. 2000. WWW-suunnittelu. Helsinki: Oy Edita Ab, IT Press.

Oksanen, T. 2010. CRM ja muutoksen tuska – Asiakkuudet haltuun. Helsinki: Talentum.

Tuomi, J. & Sarajärvi, A. 2009. Laadullinen tutkimus ja sisällönanalyysi. Helsinki: Tammi.

Vilkka, H. & Airaksinen, T. 2003. Toiminnallinen opinnäytetyö. Helsinki: Tammi.

Elektroniset lähteet

Apache HTTP Server Project 2019. What is the Apache HTTP Server Project? [viitattu 11.4.2019]. Apache HTTP Server Project. Saatavissa: https://httpd.apache.org/ABOUT_APACHE.html

Don Ho 2019. Features [viitattu 11.4.2019]. Notepad++. Saatavissa: <https://notepad-plus-plus.org/features/>

Elisa 2015. Tutkimus: Kasvuhakuiset PK-yritykset digitalisaation edelläkävijöitä. [viitattu 8.5.2019]. Elisa Oyj. Saatavissa: <https://hub.elisa.fi/kasvuhakuiset-pk-yritykset-digitalisaation-edellakavijoita/>

Fedora Project Wiki 2019. What is Extra Packages for Enterprise Linux (or EPEL)? [viitattu 11.4.2019]. Fedora Project Wiki. Saatavissa: https://fedoraproject.org/wiki/EPEL#What_is_Extra_Packages_for_Enterprise_Linux_.28or_.29_EPEL.29.3F

Kettunen, J. & Simons, M. 2001. Toiminnanohjausjärjestelmän käyttöönotto pk-yrityksessä - Teknologialähtöisestä ajattelusta kohti tiedon ja osaamisen hallintaa. Valtion teknillinen tutkimuskeskus [viitattu 8.5.2019]. Saatavissa: <https://www.vtt.fi/inf/pdf/julkaisut/2001/J854.pdf>

Laki pelastuslain muuttamisesta 5.12.2018/1078. Suomen laki [viitattu 3.4.2019].

Saatavissa: <https://www.finlex.fi/fi/laki/alkup/2018/20181078>

MariaDB Foundation 2019. About MariaDB. MariaDB Foundation [viitattu 11.4.2019].

Saatavissa: <https://mariadb.org/about/>

Pelastuslaki 29.4.2011/379. Suomen laki [viitattu 3.4.2019]. Saatavissa:

<https://www.finlex.fi/fi/laki/alkup/2011/20110379>

PHP 2019. What is PHP? PHP [viitattu 11.4.2019]. Saatavissa:

<https://www.php.net/manual/en/intro-what-is.php>

phpMyAdmin 2019. Home. phpMyAdmin [viitattu 11.4.2019]. Saatavissa:

<https://www.phpmyadmin.net/>

Royce, W. 1970. Managing the Development of Large Software Systems [viitattu

8.5.2019]. Saatavissa: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>

Terho, K. 2016. Tietojärjestelmän vaatimusmäärittelyt ja valintaperusteet pienelle liikuntakeskukselle. Tampereen yliopisto. Pro gradu -tutkielma [viitattu 8.5.2019].

Saatavissa: <https://tampub.uta.fi/bitstream/handle/10024/100353/GRADU-1481893824.pdf?sequence=1&isAllowed=y>

Tietosuojavaltuutetun toimisto 2019. Henkilötietojen käsittely. Tietosuojavaltuutetun

toimisto [viitattu 23.3.2019]. Saatavissa: <https://tietosuoja.fi/henkilotietojen-kasittely>

Tuisku, M. & Niemelä, S. 2017. Tietojärjestelmä asiakastietojen sähköiseen kirjaamiseen.

Oulun ammattikorkeakoulu. YAMK-opinnäytetyö [viitattu 8.5.2019]. Saatavissa:

https://www.theseus.fi/bitstream/handle/10024/125864/Tuisku_Maiju%20ja%20Niemela_Sakari.pdf?sequence=1&isAllowed=y

w3schools 2019a. HTML Introduction. w3schools [viitattu 11.4.2019]. Saatavissa:

https://www.w3schools.com/html/html_intro.asp

w3schools 2019b. HTML5 Introduction. w3schools [viitattu 2.5.2019]. Saatavissa:

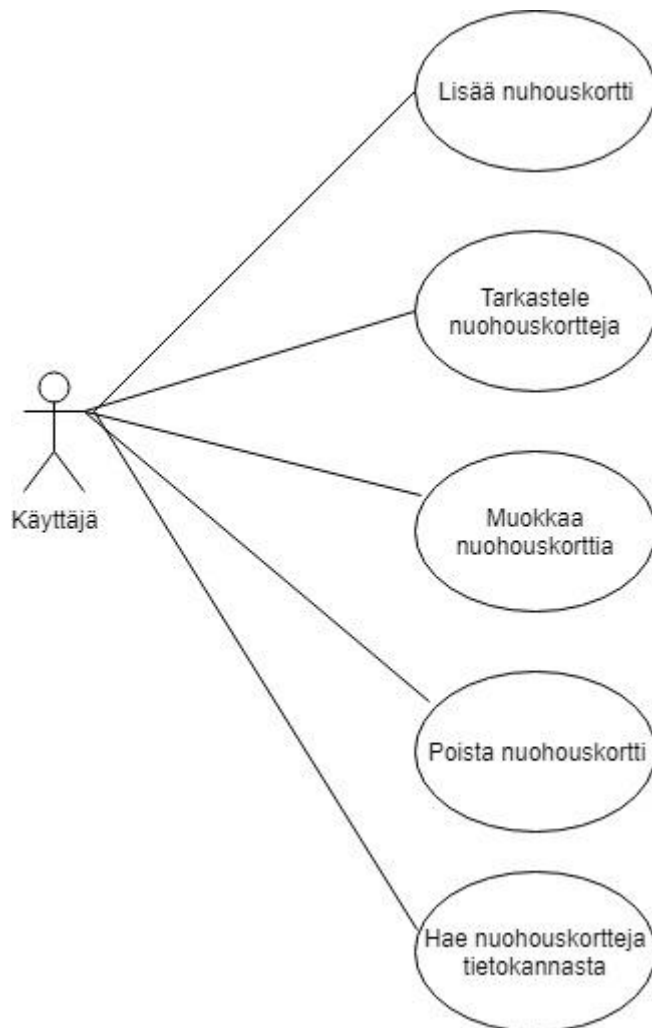
https://www.w3schools.com/html/html5_intro.asp

w3schools 2019c. CSS Introduction. w3schools [viitattu 11.4.2019]. Saatavissa:

https://www.w3schools.com/csS/css_intro.asp

LIITTEET

Liite 1. Käyttötapauskaavio



Liite 2. Käyttötapauksien kuvaaminen

#1 Lisää uusi nuohouskortti

Nimi:	Lisää nuohouskortti
Toimija:	Käyttäjä
Kuvaus	Tässä käyttötapauksessa käyttäjä lisää nuohouskortin järjestelmään.
Ennakko-vaatimukset	Käyttäjä on kirjautunut sisään
Tapahtumien kulku	1. Käyttäjä klikkaa Syötä uusi nuohouskortti -painiketta 2. Selainäkymään aukeaa tekstikenttiä ja valikoita 3. Käyttäjä täyttää kaikki tarpeelliset kentät 4. Käyttäjä klikkaa Tallenna-painiketta 5. Verkkosivulla näkyy ilmoitus: "Nuohouskortti tallennettu"
Lopputulos	Nuohouskorttiin täytetyt tiedot on tallennettu tietokantaan
Vaihtoehtoiset tapahtumat	1. Pakollisia kenttiä ei ole täytetty. Verkkosivulla näkyy ilmoitus: "Täytä pakolliset kentät"
Muuta	-

#2 Tarkastele nuohouskortteja

Nimi:	Tarkastele nuohouskortteja
Toimija:	Käyttäjä
Kuvaus	Tässä käyttötapauksessa käyttäjä tarkastelee tallennettuja nuohouskortteja.
Ennakko-vaatimukset	Käyttäjä on kirjautunut sisään
Tapahtumien kulku	1. Käyttäjä klikkaa Tarkastele ja muokkaa nuohouskortteja -painiketta 2. Selainäkymään aukeaa tallennettuja nuohouskortteja niiden täyttämisyjärjestyksessä
Lopputulos	Käyttäjä voi tarkastella järjestelmään tallennettuja nuohouskortteja
Vaihtoehtoiset tapahtumat	-
Muuta	Liittyy käyttötapauksiin #3 ja #4

#3 Muokkaa nuohouskorttia

Nimi:	Muokkaa nuohouskorttia
Toimija:	Käyttäjä
Kuvaus	Tässä käyttötapauksessa käyttäjä muokkaa tallennettua nuohouskorttia.
Ennakko-vaatimukset	Käyttäjä on kirjautunut sisään Käyttäjä on klikannut Tarkastele ja muokkaa nuohouskortteja -painiketta
Tapahtumien kulku	1. Käyttäjä on Tarkastele ja muokkaa nuohouskortteja näkymässä 2. Tallennetut nuohouskortit näkyvät selaimessa 3. Käyttäjä klikkaa Muokkaa-painiketta haluamansa kortin kohdalla 4. Selain avaa halutun kortin muokkausnäkyssä 5. Käyttäjä muokkaa haluamansa kentät 6. Käyttäjä painaa sivun alalaidan Tallenna-painiketta 7. Verkkosivulla näkyy ilmoitus: ”Muutokset tallennettu”
Lopputulokset	Käyttäjä on muokannut nuohouskortin tietoja
Vaihtoehtoiset tapahtumat	1. Pakollisia kenttiä ei ole täytetty. Ruudulla näkyy ilmoitus: ”Täytä pakolliset kentät”
Muuta	Liittyy käyttötapaukseen #2

#4 Poista nuohouskortti

Nimi:	Poista nuohouskortti
Toimija:	Käyttäjä
Kuvaus	Tässä käyttötapauksessa käyttäjä poistaa tallennetun nuohouskortin.
Ennakko-vaatimukset	Käyttäjä on kirjautunut sisään Käyttäjä on klikannut Tarkastele ja muokkaa nuohouskortteja -painiketta
Tapahtumien kulku	1. Käyttäjä on Tarkastele ja muokkaa nuohouskortteja -näkyssä 2. Tallennetut nuohouskortit näkyvät selaimessa 3. Käyttäjä klikkaa Poista-painiketta haluamansa kortin kohdalla 4. Selain avaa ikkunan: ”Haluatko varmasti poistaa tämän kortin?” 5. Käyttäjä klikkaa OK-painiketta
Lopputulokset	Haluttu nuohouskortti poistuu järjestelmästä
Vaihtoehtoiset tapahtumat	1. Käyttäjä klikkaa Peruuta-painiketta 2. Nuohouskortti ei poistu järjestelmästä
Muuta	Liittyy käyttötapaukseen #2

#5 Hae nuohouskortti

Nimi:	Hae nuohouskortti
Toimija:	Käyttäjä
Kuvaus	Tässä käyttötapauksessa käyttäjä hakee nuohouskortteja.
Ennakkovaatimukset	Käyttäjä on kirjautunut sisään
Tapahtumien kulku	<ol style="list-style-type: none"> 1. Käyttäjä klikkaa Hae-painiketta 2. Selain avaa Hae nuohouskortteja -näkyvän 3. Käyttäjä klikkaa Syötä hakusana -kenttää 4. Käyttäjä syöttää hakusanan 5. Käyttäjä klikkaa Hae-painiketta
Lopputulos	Ruudulla näkyvät syötetyn hakusanan sisältävät kortit
Vaihtoehtoiset tapaukset	1. Hakusanalla ei löydy tuloksia
Muuta	