LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Sähköinen varausjärjestelmä Laurean Studio Oivalle

Jokinen, Tommi

2016 Laurea

Laurea-ammattikorkeakoulu

Sähköinen varausjärjestelmä Laurean Studio Oivalle

Tommi Jokinen
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Lokakuu, 2016

Laurea-ammattikorkeakoulu **Tiivistelmä**
Tietojenkäsittelyn koulutusohjelma
Tradenomi (AMK)

Tommi Jokinen

Sähköinen varausjärjestelmä Laurean Studio Oivalle

Vuosi 2016 Sivumäärä 39

Opinnäytetyön aiheena oli tehdä Laurea-ammattikorkeakoulussa toimivalle Studio Oivalle sähköinen varausjärjestelmä. Studio Oiva lainaa kalustoaan Laurean opiskelijoille, ja aikaisemmin se tapahtui paperisen varausjärjestelmän avulla.

Opinnäytetyössä tutkittiin käyttäjäystävällistä suunnittelua, käytettävyyttä ja käyttökokemusta, sekä siinä tutustuttiin lyhyesti yleisimpiin ohjelmistokehityksen vaihejakomalleihin. Projektissa käytettyjä ohjelmointikieliä käytiin myös lyhyesti läpi.

Työn aikana suoritettiin havainnointia, jonka avulla saatiin tietoa varausjärjestelmän käyttäjien tarpeista. Vanha toimintatapa kuvattiin myös toimintatarinalla, joka havaintojen kanssa antoi hyvän pohjan uuden järjestelmän suunnittelulle. Uusi toimintatapa kuvattiin käyttötarinalla, ja se auttoi uuden järjestelmän suunnitelman testaamisessa.

Opinnäytetyön lopputuloksena syntyi verkossa pyörivä sähköinen varausjärjestelmä, joka yksinkertaisti ja helpotti varausprosessia. Työssä kerrotaan varausjärjestelmän toiminnallisuuksista ja ohjelmistoista, jotka olivat toteutusvaiheessa tärkeimpiä.

Asiasanat: Verkkosivut, Varausjärjestelmä, Tietokannat, PHP

Laurea University of Applied Sciences **Abstract**
Degree Programme in Business Information Technology
Bachelor's Thesis

Tommi Jokinen

Electronic reservation system for Laurea's Studio Oiva

Year 2016 Pages 39

The aim of this thesis was to create an electronic reservation system for Studio Oiva. Studio Oiva operates at Laurea University of Applied Sciences. Studio Oiva loans equipment to students at Laurea and previously it had been done using a paper based reservation system.

This thesis explores user friendly design, usability and user experience. Most common software development models are also briefly examined along with the programming languages that were used in making the reservation system.

In process of making this thesis users of the old loaning service were observed to gain information on their needs. Loaning service's old operating model was also depicted with a story to gain information on its flaws. Gained information was used to produce a plan for the new reservation system. Operating model of the planned reservation system was depicted with a story to help test it.

As a result, this thesis produced a new reservation system that operates online. New system simplified the reservation process and made it easier to manage. Functionalities of the new system are explored in this thesis along with the programs that were used in making it.

Keywords: Websites, Reservation system, Databases, PHP

Sisällys

1 Johdanto

Opinnäytetyön tarkoitus oli toteuttaa käyttäjäystävällinen varausjärjestelmä toimeksiantajalle, ja tätä tarkoitusta varten työssä tutustutaan lyhyesti käytettävyyteen ja käyttökokemukseen lyhyesti käsitteinä. Työssä tutustutaan myös verkkokehitykseen ja verkkosivujen suunnitteluun, jotta verkkosivun toteuttamiseen saadaan tietopohjaa. Varausjärjestelmän toteuttamista varten työssä tutustutaan yleisimpiin ohjelmistokehityksen vaihejakomalleihin, vesiputousmalliin sekä ketteriin menetelmiin. Vaihejakomallien tarkoitus oli tuoda selkeää rakennetta varausjärjestelmän toteutukseen. Työssä päädyttiin soveltamaan näitten kahden mallin sekoitusta.

Varsinainen varausjärjestelmän toteutus alkoi vaatimusmäärittelyllä, jonka jälkeen lainauspalvelun käyttäjiä alettiin havainnoida. Toimin havainnoinnissa varauspalvelua pyörittävänä palveluhenkilönä. Havainnoinnin tarkoitus oli saada tietoa käyttäjien tarpeista ja toimintatavoista, jotta sähköinen varausjärjestelmä voitaisiin toteuttaa mahdollisimman käyttäjäystävälliseksi.

Opinnäytetyössä käytetään opiskelijalta saatua toimintatarinaa, jolla kuvattiin opiskelijan toimintatapaa käytettäessä vanhaa järjestelmää. Toimintatarinan tarkoitus oli auttaa suunnittelijaa löytämään vanhan toimintatavan puutteita, jotta niitä voitaisiin korjata uudella järjestelmällä. Toimintatarinan lisäksi työssä tehtiin käyttötarina, joka rakennettiin toimintatarinan pohjalta. Käyttötarinalla kuvattiin opiskelijan toimintatapaa uudessa lainauspalvelussa, ja sen tarkoitus oli testata uuden järjestelmän suunnitelmaa.

Tarinoitten jälkeen alkoi sivujen toiminnallisuuksien suunnittelu prototypoinnilla. Prototypointi aloitettiin yksinkertaisista paperiprototyypeistä, joita voitiin iteroida nopeasti. Iterointikierroksen päätteeksi prototyyppejä testattiin käyttäjällä. Hyvän toimivaksi todetun prototyypin valmistuttua, oli aika suunnitella sivuston ulkonäkö. Sivuston ulkonäkö suunniteltiin rautalankamalleja toteuttamalla.

Opinnäytetyössä käydään läpi tietokantapohjaisen verkkopalvelun luomiseen tarvittavia osia, sekä niitten toteuttamisessa käytettyjä ohjelmistoja. Verkkopalvelun osat koostuvat verkkosivuista jotka tehtiin opinnäytetyössä HTML-merkintäkielellä ja CSS-tyylikielellä, sekä varsinaisesta varausjärjestelmästä, joka tehtiin SQL-kyselykielellä ja PHP-ohjelmointikielellä. Projektissa käytettiin XAMPP-palvelinpakettia, jonka avulla voitiin luoda paikallinen kehitysympäristö projektin kehittämiseen. Verkkosivut toteutettiin Adobe Dreamweaver -ohjelmistolla, josta löytyy tähän hyödyllisiä toimintoja kuten koodin katselmointi ja täydennys. Projektissa käytetyt tietokannat luotiin XAMPP-palvelinpaketin sisältämään MariaDB-relaatiotietokantahallintajärjestelmään.

Opinnäytetyön lopussa kerrotaan työn tuotoksesta, sähköisestä varausjärjestelmästä. Järjestelmän tärkeimmät toiminnot, kuten varaustoiminto, sekä varauksien hallinta käydään läpi. Sivuston rakenne sekä järjestelmän tietokantarakenne kuvataan myös lopuksi.

2 Lähtökohdat

Opinnäytetyön toimeksiantaja oli Laurea Ammattikorkeakoulussa toimiva audiovisuaalinen palveluyksikkö Studio Oiva, joka tarvitsi sähköisen varausjärjestelmän. Studio Oiva tarjoaa Laurean henkilöstölle sekä oppilaille mahdollisuuden lainata kalustoa Oivasta, kuten videokameroita ja sanelukoneita. Aikaisemmin Studio Oivalla oli käytössä paperinen varausjärjestelmä. Vanha varausjärjestelmä koostui useista A4 varauslistoista yhdessä mapissa, tehden varauksien tekemisen mahdottomaksi muualta kuin Studio Oivan toimistosta. Sähköisen varausjärjestelmän tavoite on yksinkertaistaa varausprosessia, sekä tehdä varauksien hallinnasta helpompaa.

Opinnäytetyön aikana olin työharjoittelussa toimeksiantajalla, ja työtehtäviini kuului vastata paperisesta varausjärjestelmästä. Tämä antoi mahdollisuuden havainnoida varauspalvelun käyttäjiä, ja oppia heidän käyttäytymistavoista ja tarpeista. Havaintojen perusteella saavutettiin tietopohja, joka auttoi varausjärjestelmän suunnittelussa.

2.1 Työn rajaukset

Tutkimuksellinen tavoite opinnäytetyössä on tutkia käyttäjäystävällistä suunnittelua, jonka avulla on tarkoitus toteuttaa käyttäjäystävällinen varausjärjestelmä. Opinnäytetyössä tutkitaan verkkokehitystä yleisesti, jotta saavutetaan riittävä tietopohja järjestelmän kehittämiselle. Työssä tutkitaan myös lyhyesti ohjelmistokehityksen vaihejakomalleja vesiputousmallia ja ketteriä menetelmiä, sekä projektissa käytettyjä HTML-, CSS-, SQL- ja PHP-kieliä.

Opinnäytetyön aiheesta on rajattu pois sisällönhallintajärjestelmät kuten Drupal ja WordPress. Muut ohjelmointikielet kuten HTML, CSS, SQL ja PHP sähköisen varausjärjestelmän toteutustapoina ovat myös rajattu pois. Opinnäytetyössä ei myöskään tutkita verkkosivujen mobiililaitteille optimointia, eikä mobiiliversioiden kehittämistä.

2.2 Käsiteluettelo

HTML eli Hyper Text Markup Language on merkintäkieli, jolla kuvaillaan verkkosivujen rakennetta. HTML-sivut koostuvat elementeistä joita esitetään merkinnöillä. Selaimet lukevat HTML-kieltä ja tulkkaavat sivujen merkinnöt käyttäjälle. (W3Schools 2016.)

PHP eli Hypertext Preprocessor on avoimen lähdekoodin ohjelmointikieli mikä soveltuu erityisesti verkkokehitykseen. PHP-ohjelmakoodia voidaan sisällyttää HTML-dokumenttiin. Ohjelmakoodi suoritetaan palvelimen puolella, eikä käyttäjän selaimessa kuten esimerkiksi JavaScriptiä. (PHP 2016.)

CSS eli Cascading Style Sheets on tyylikieli, jonka avulla kuvaillaan, miten HTML-elementtejä esitetään selaimessa. CSS-tyylikielen avulla voidaan säästää aikaa määrittelemällä elementeille sääntöjä, joitten perusteella elementit esitetään. Tyylikieltä voidaan käyttää HTML-dokumentin sisällä, vaikkakin sitä yleensä käytetään erillisessä tiedostossa, johon viitataan HTML-dokumentissa. (W3Schools 2016.)

SQL eli Structured Query Language on kieli, jonka avulla käsitellään tietokantoja. SQL on ANSI (American National Standards Institute) mukaan standardi relaatiotietokantahallintajärjestelmille. SQL-kielessä toimintoja kuten tiedon hakemista tai tiedon päivittämistä suoritetaan lausunnoilla. (SQLCourse 2016.)

Avoin lähdekoodi tarkoittaa tapaa tuottaa ja jakaa ohjelmistoa. Avoimen lähdekoodin ohjelmistojen tulee sisältää ohjelman lähdekoodi, jota kuka tahansa voi muokata ja jakaa vapaasti. Joissakin tapauksissa ohjelmistot eivät automaattisesti sisällä lähdekoodia, mutta käyttäjällä tulee olla tapa hankkia lähdekoodi avoimen lähdekoodin ohjelmistoista. (Open Source Initiative 2016.)

Relaatiotietokanta on tietokantamalli, joka koostuu taulukoista. Relaatiotietokannassa tiedot tallennetaan taulukkojen riveille, ja saman tiedon tallentamista useaan eri taulukkoon pyritään välttää. Relaatiotietokannoissa taulukkojen välille luodaan yhteyksiä joka nopeuttaa tiedon etsimistä. (Jari Sarja 2006.)

3 Verkkokehitys

Käyttäjäkeskeisillä suunnittelumenetelmillä pyritään hyvään käytettävyyteen, käyttäjäkokemukseen ja tehokkuuteen. Menetelmissä otetaan huomioon käyttäjäryhmät, heidän tarpeensa, sekä missä ja miten he toimivat. Käyttäjäkeskeisistä menetelmistä hyötyvät hyvän suunnittelutyön takia käyttäjät, mutta myös itse suunnittelijat. Suunnittelijat saavat tietoa käyttäjän maailmasta tutkimuksen perusteella, ja tutkimustyöllä taataan suunnittelutyön oikeaan suuntaan vieminen. Käyttötarinoiden ja kuvakertomuksien avulla saadaan ymmärrystä käyttäjäkokemuksesta ja niiden avulla voidaan pohtia tuotteen toteutustapaa. Prototyyppejä tekemällä saavutetaan perusta testaukselle ja arvioinnille, jo ennen varsinaista tuotteen toteutusta. (Sinkkonen, Nuutila & Törmä 2009, 27.)

Sivustoa suunniteltaessa tulisi ottaa huomioon myös muut käyttöjärjestelmät ja selaimet. Järjestelmäriippumattomuus verkkosivujen yhteydessä tarkoittaa, että niitä voidaan käyttää myös toisilla selaimilla ja käyttöjärjestelmillä. Nykypäivänä kuitenkin verkkosivujen toimivuusongelmat eri selainten välillä ovat harvinaisia standardien ansiosta. (Nielsen & Loranger 2006, 94-95.)

Mobiililaitteille suunniteltaessa on usein tarpeellista tehdä erillinen mobiiliversio verkkosivustosta perinteisen tietokoneelle tarkoitetun version ohelle. Vaikka perinteinen pöytäkoneille tarkoitettu versio toimisikin mobiililaitteilla, se tuskin antaa hyvää käyttäjäkokemusta kummankin käyttäjäryhmille. Esimerkiksi Jo pelkästään mobiililaitteiden ruutujen kokoero verrattuna perinteisiin näyttöihin on yleensä riittävä perustelu erillisen mobiiliversion toteuttamiselle. (Nielsen & Loranger 2006, 96.)

3.1 Käytettävyys ja käyttökokemus

Käytettävyydellä tarkoitetaan palvelun käyttölaatua ja käyttäjäkokemuksella kokemuksen laatua. Käyttäjäkokemus ja käytettävyys kulkevat käsi kädessä. Käyttäjäkokemuksen ollessa hyvä, käyttäjä todennäköisesti on valmis sietämään virheitä verkkosivuston käytettävyydessä. Käyttäjäkokemuksen ollessa huono, sivuston virheet korostuvat huonona käytettävyytenä. (Sinkkonen ym. 2009, 18-19.)

Monien yrityksien verkkopalvelujen pahimpia kilpailijoita ovat lukemattomat toiset verkkopalvelut, ja toisten taas puhelimet. Puhelimien kustannukset yhteydenpitovälineenä yrityksen asiakkaisiin ovat yleensä yrityksille suuremmat kuin heidän verkkopalveluiden. Verkkopalvelu tulisikin tehdä tarpeeksi helppokäyttöiseksi ja houkuttelevaksi asiakkaalle, jotta he eivät turhaudu heti palveluun, ja turvaudu palvelun sijasta soittamaan asiakaspalveluun. (Sinkkonen ym. 2009, 17.)

3.2 Verkkosivujen suunnittelu

Verkkosivuille tulee luoda selkeä visuaalinen hierarkia, joka viestii käyttäjälle visuaalisilla vihjeillä selkeästi sivujen osien välisiä suhteita, eli mitkä osat sivusta kuuluvat samaan kategoriaan ja mitkä kuuluvat alakategorioihin. Verkkosivujen suunnittelussa on hyvä sijoittaa sivujen tärkeimmät osat selkeästi näkyvälle, antaa niille riittävästi tilaa, sekä mielellään käyttää myös suurempaa fonttikokoa, jotta ne erottuvat vähemmän tärkeämmistä osista. Hyvällä visuaalisella hierarkialla on mahdollista säästää käyttäjän aikaa ja vaivaa sivujen tulkinnassa. (Krug, 2006, 31-33.)

Visuaalista kohinaa, eli verkkosivujen ymmärtämistä haittaavaa sekavuutta tai taustakohinaa on hyvä välttää verkkosivuja suunniteltaessa. Sekavuus voi esiintyä esimerkiksi usean kirkkaan

tekstin muodossa, jotka ”hyppäävät” esiin, samalla kilpaillen toistensa kanssa käyttäjän huomiosta. Taustakohina taas tarkoittaa useaa häiriötekijää, jotka eivät yksinään ole käyttäjää uuvuttavia tekijöitä, mutta kokonaisuutena kyllä. Käyttäjien sietokyky kohinaa kohtaan vaihtelee, mutta verkkosivujen suunnittelijan näkökulmasta kaikkea kohinaa tulee välttää. (Krug, 2006, 38-39.)

3.3 Toimintatarina ja käyttötarina

Tarinat ovat tapa kuvata persoonien toimintatapaa tiivistetysti, ja niissä käytetään persoonia kuvaamaan käyttäjäryhmiä tiivistetysti. Tarinoitten avulla täydennetään suunnittelijan tietämystä palvelusta, ja tarinoitten paikkansapitävyys kannattaakin tarkistaa käyttäjiltä, jotta mahdolliset aukot tarinassa voidaan korjata. Tarina kertoo aina yhden persoonan toimintatavasta. (Sinkkonen ym. 2009, 135-136.)

Toimintatarina on tapa kuvata vanhaa virheellistä tai puutteellista toimintatapaa. Toimintatarinan tavoite on auttaa suunnittelijaa hahmottamaan käyttäjän pyrkimyksiä sekä ongelmia toimintatavassa. Toimintatarina voidaan tehdä käyttäjätutkimuksen perusteella, taikka ne voidaan pyytää käyttäjältä suoraan. (Sinkkonen ym. 2009, 135-136.)

Käyttötarina on tapa kuvata persoonien toimintatapaa verkkopalvelussa, ja sen tarkoitus on suunnitelman testaus ja ilmaisu. Käyttötarinoissa on hyvä kuvata tarkasti persoonan toiminta verkkopalvelussa, ja tarinaa täydentämään on myös hyvä tehdä itse persoonan esittely. Käyttötarina tehdään yleensä toimintatarinoiden sekä kaiken muun kerätyn materiaalin pohjalta. Jos ollaan parantamassa vanhaa palvelua uuden tekemisen sijasta, käyttötarinassa panostetaan yleensä toimintatarinoiden ongelmakohtiin. (Sinkkonen ym. 2009, 171-172.)

4 Ohjelmistokehityksen vaihejakomallit

Ohjelmistokehityksessä on useita vaiheita, jotka muodossa tai toisessa toteutuvat aina. Vaiheisiin kuuluvat:

- Vaatimusmäärittely
- Suunnittelu
- Toteutus
- Testaus
- Julkaisu
- Ylläpito/jatkokehitys

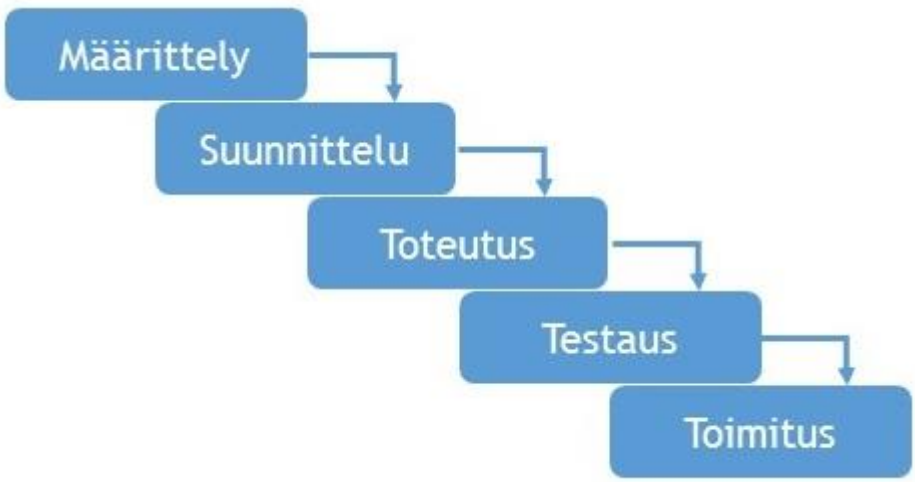
Vaiheitten toteutumisjärjestys vaihtelee ohjelmistonkehityksessä käytettävän vaihejakomallin mukaan. (Dooley 2011, 7.)

Yleisesti vaihejakomallit ovat joko suunnitelmaohjautuvia taikka ketteriä menetelmiä. Suunnitelmaohjautuvissa lähestymistavoissa vaiheet käydään järjestyksessä läpi. Ketterissä lähestymistavoissa vaiheita yleensä iteroidaan useasti. (Dooley 2011, 8.) Näitä kahta vaihejakomallia avataan seuraavissa alaluvuissa.

4.1 Vesiputousmalli

Vesiputousmalli on perinteinen suunnitelmaohjautuva vaihejakomalli ja se vaatii tarkkaa dokumentointia sen jokaisessa vaiheessa (Dooley 2011, 9). Vesiputousmalli on lineaarinen vaihejakomalli, jossa käydään sen vaiheet aina järjestyksessä läpi, suorittaen aina aikaisempi vaihe valmiiksi ennen kuin seuraavaa vaihetta voidaan aloittaa. Malli perustuu laadukkaaseen suunnitteluun, koska vaiheitten muokkaaminen niitten valmistumisen jälkeen käytännössä tarkoittaa työn uudelleen tekemistä. Mallissa jokainen vaihe dokumentoidaan tarkasti, ja se tarkan suunnittelun kanssa mahdollistaa työntekijöiden vaihdokset kesken projektin. (Vilmunen 2015.)

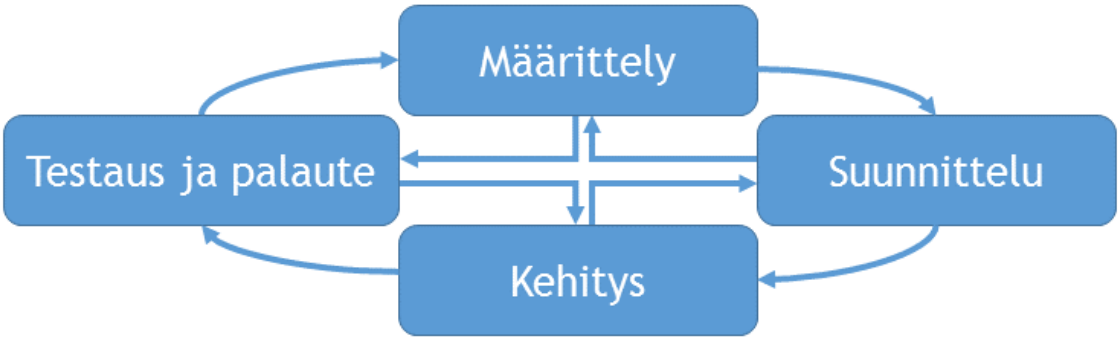
Menetelmän selkeä vahvuus on siinä tapahtuvan määrittelyn ja suunnittelun laajuus, jonka perusteella asiakas tietää projektin aikataulun, kustannukset sekä mitä odottaa lopputulokselta. Heikkous mallissa on sen lineaarisuus, jonka takia aikaisempiin vaiheisiin palaaminen voi tarkoittaa isoja kustannuksia. (Vilmunen 2015.) Kuviossa 1 on vesiputousmallin tyypilliset vaiheet.



Kuvio 1: Vesiputousmallin vaiheet

4.2 Ketterät menetelmät

Ketterät menetelmät ovat iteratiivinen suuntaus ohjelmistokehitykseen, jossa työ jaetaan kehitysjaksoihin. Kehitysjaksot koostuvat listasta toimituskelpoisia osia, jotka toteutetaan asiakkaan määrittämän tärkeysjärjestyksen perusteella. Tehdyt työt esitellään kehitysjakson päätteeksi, jolloin asiakkaalla ja projektiryhmällä on mahdollisuus arvioida tehtyä työtä. Asiakas saa tällöin mahdollisuuden ohjata projektia palautteen avulla. Ketterien menetelmien projekteissa voidaan helpommin kiirehtiä tuotteen julkaisua kuin vesiputousmallia noudattavissa projekteissa. Julkaisukelpoisen version tuottamista voidaan kiirehtiä jättämällä vähemmän tärkeitten toimintojen kehittäminen vasta julkaisun jälkeen. (Lotz 2013.) Kuviossa 2 näkyy useasti iteroitavat vaiheet.



Kuvio 2: Ketterien menetelmien vaiheet

Asiakas on tärkeässä osassa ketteriä menetelmiä, ja menetelmien periaate on asiakkaan tyytyväisyyden takaaminen. Tyytyväisyys taataan asiakkaan muuttuvien tarpeitten huomioon ottamisella, sekä jatkuvalla yhteistyöllä asiakkaan kanssa. Ketterissä menetelmissä kehitystiimien tulisi pohtia säännöllisesti, miten tiimin tehokkuutta voitaisiin lisätä. (Agile Manifesto 2001.)

5 Tutkimus

Tässä luvussa käydään läpi lyhyesti kvalitatiivista ja kvantitatiivista tutkimusta, sekä toimintatutkimusta. Luvussa kerrotaan myös tutkimusta tehtäessä huomioon otettavista aiheista kuten lähdekritiikistä, reliabiliteetista ja validiteetista. Aineistonkeruu menetelmää havainnointia avataan myös luvussa.

Tutkimuksen avulla voidaan tuottaa uutta tietoa sellaisten ongelmien ratkaisuun, jotka eivät välttämättä jokapäiväisen arkiajattelun perusteella ratkea. Uusi tieto auttaa ymmärtämään

ongelmia paremmin, sekä sen avulla löydetään uusia keinoja ongelmista selviämiseen (Hirsjärvi, Remes & Sajavaara 2007, 19.) Tämä tutkimus on toiminnallinen toimintatutkimus. Toimintatutkimuksen tuotoksena toimeksiantaja saa uuden sähköisen varausjärjestelmän.

5.1 Kvalitatiivinen ja kvantitatiivinen tutkimus

Kvalitatiivisella tutkimuksella, eli laadullisella tutkimuksella pyritään ymmärtämään ilmiötä kokonaisvaltaisesti, sen laadun, ominaisuuksien, sekä merkityksien pohjalta. Tutkimustyylissä voidaan hyödyntää useita erilaisia menetelmiä. Menetelmien yhteisinä piirteinä esiintyy usein esimerkiksi ilmiön esiintymisympäristöön ja taustaan liittyvät näkökulmat. (Jyväskylän yliopisto 2015.)

Kvalitatiivisen tutkimuksen lähtökohtana pidetään todellisuuden kuvaamista. Tutkimuksessa pitää kuitenkin muistaa, että todellisuutta ei voi jakaa harkitsemattomasti useaan eri osaan. Osat muovaavat toinen toisiaan ja tutkimuksessa onkin punnittava osien merkitystä toisiinsa. Kvalitatiivisten tutkimuksien tavoitteena pidetään yleensä totuuksien löytämistä tai paljastamista, olemassa olevien väittämien todentamisen sijasta. (Hirsjärvi ym. 2007, 157.)

Kvantitatiivisella tutkimuksella, eli määrällisellä tutkimuksella pyritään ymmärtämään ilmiötä tilastojen ja numeroiden tulkinnan avulla. Määrällisessä tutkimuksessa usein keskitytään ilmiön luokitteluun, syy- ja seuraussuhteisiin, vertailuun, sekä numeeristen tulosten selittämiseen. Vaikka kvantitatiivista ja kvalitatiivista tutkimustyyppiä pidetäänkin hyvin erilaisina, voidaan niitä kuitenkin käyttää samassa tutkimuksessa saman ilmiön selittämiseen. Koska tutkimustyyppien useat menetelmät ovat hyvin erilaisia, ne voivat selittää samoja ilmiötä eri tavoin. (Jyväskylän yliopisto 2015.)

Sekä kvalitatiivisesta että kvantitatiivisesta tutkimuksesta löytyy samankaltaisuuksia. Kummassakin tutkimustyypissä haastatteluja voidaan käyttää tiedonkeruun muotona. Näitten tutkimusmenetelmien haastatteluissa on kuitenkin erilaisuuksia, kuten niitten kohderyhmät. Määrällisessä tutkimuksessa haastateltavat valitaan yleensä satunnaisesti, kun taas laadullisessa tutkimuksessa haastateltavat ovat tutkijan valitsemia. Eroja löytyy myös haastattelujen rakenteesta. Määrällisen tutkimuksen haastatteluissa haastattelija käyttää ennalta valittuja kysymyksiä, kun taas laadullisessa tutkimuksessa käytetty haastattelun perusmuoto on avoin kysymysaihe. (Tilastokeskus 2016.)

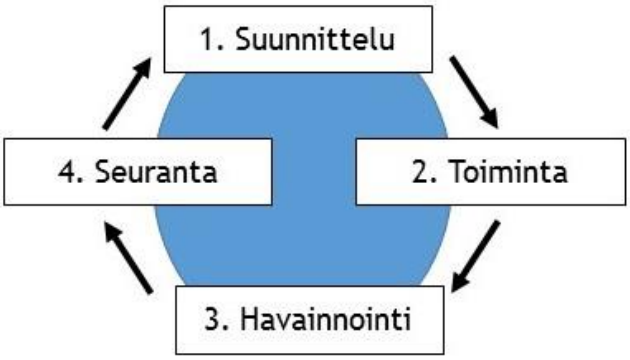
Kummassakin tutkimustyypissä pyritään objektiivisuuteen, mutta objektiivisuuden lähtökohta on eri. Määrällisessä tutkimuksessa haastattelija pysyttäytyy erillään haastateltavasta, haastattelussa tapahtuvia kysymyksiä lukuun ottamatta. Laadullisessa tutkimuksessa haastattelija taas yrittää ymmärtää haastateltavan näkökulmia ja asenteita mahdollisimman hyvin pyrkimällä vuorovaikutukseen haastateltavan kanssa, sekoittamatta kuitenkaan omia uskomuksiaan

ja asenteitaan haastatteluun. Tutkimuksien johtopäätöksien ilmaisuissa on myös eroja. Laadullisessa tutkimuksessa tulokset ovat usein tekstiä, kun taas määrällisessä tutkimuksessa tulokset esitetään tilastoina sekä kaavioina. (Tilastokeskus 2016.)

5.2 Toimintatutkimus

Toimintatutkimuksen tutkimusprosessi on syklinen, jossa yksinkertaisimmillaan sen sykliin kuuluvat suunnittelu, toiminta ja seuranta. Toimintatutkimuksen määrittäminen tutkimusmenetelmänä on vaikeaa, koska se on sekoitus kvalitatiivisia ja kvantitatiivisia tutkimusmenetelmiä. Tutkimusmenetelmänä se on hyvin lähellä kehittämistutkimusta, mutta niitten metodologioissa on kuitenkin selvä ero. (Kananen 2014, 13-14.) Toimintatutkimuksen tavoite on muutos, johon päästään ongelman ratkonnan kautta. Tutkimuksessa tutkija osallistuu tutkittavan ilmiön toimintaan ja tutkittavaan yhteisöön. (Kananen 2014, 28.)

Tässä opinnäytetyössä kuvion 3 vaiheitten toteutumista voidaan kuvata seuraavanlaisesti. Suunnitteluvaiheessa tutkitaan verkkosivujen teoriaa sekä suoritetaan käyttäjätutkimusta. Toimintavaiheessa suunnitellaan ja toteutetaan verkkosivustoa. Havainnointivaiheessa tapahtuu sivujen testaus ja käyttöönotto. Seurantavaiheessa taataan sivuston toimivuus sivujen ylläpidolla, sekä seurataan sen tuomia vaikutuksia.



Kuvio 3: Toimintatutkimuksen sykli

5.3 Lähdekritiikki

Lähdekritiikillä tarkoitetaan lähteen kelpoisuuden arvioimista. Lähteen kelpoisuutta tulisi arvioida lähteen valinta vaiheen lisäksi myös sitä luettaessa. Vaikka lähde vaikuttaisikin kelvolliselta ensisilmäyksellä, on hyvin mahdollista, että lähteen tarjoama tieto ei olekaan tarkoitusta varten sopivaa. (Hirsjärvi ym. 2007, 109)

Lähteitä valittaessa tutkijalta vaaditaan kriittisyyttä, sillä niitten kelvollisuuteen vaikuttaa monta asiaa. Tutkijan tulisi punnitta lähteen kirjoittajan uskottavuutta alalla, sillä uuden kirjailijan teksti ei välttämättä ole yhtä luotettavaa kuin esimerkiksi kirjailijan, jolla on jo useita kirjoja kirjoitettuna alalta. Tutkijan on pyrittävä käyttämään mahdollisimman tuoreita

lähteitä välttääkseen mahdollisesti vanhentuneen tiedon hankkimista. Lähteen puolueettomuutta tulisi myös punnita sekä kuinka hyvin ne kuvaavat todellisuutta. Tekstin tarkoitusta on myös hyvä pohtia, kenelle sekä mitä varten teksti on kirjoitettu. (Hirsjärvi ym. 2007, 109-110.)

5.4 Reliabiliteetti ja validiteetti

Tutkimuksen validiteetilla tarkoitetaan tutkimuksen pätevyyttä. Tutkimuksen pätevyys koostuu useasta osasta, onko tutkimus tehty huolellisesti, ja onko siinä tehdyt päätelmät ”oikeita”. Laadullisessa tutkimuksessa, joka ei kuvaa totuutta, voidaan validiteettia pitää enemmänkin tutkimuksen uskottavuutena ja vakuuttavuutena. (Saaranen-Kauppinen & Puusniekka 2006.)

Tutkimuksen reliabiliteetilla tarkoitetaan tutkimuksen luotettavuutta, ajallista luotettavuutta sekä tulosten johdonmukaisuutta. Esimerkiksi jos mittaustulokset, tai havainnot vaihtelevat eri mittauskerroilla, niin tuloksien luotettavuutta pitää kyseenalaistaa. Ongelmalliseksi mittaustulosten arvioimisen tekee se, että laadullisessa tutkimuksessa esimerkiksi harvoin tutkitaan muuttumattomia kohteita. (Saaranen-Kauppinen & Puusniekka 2006.)

5.5 Havainnointi aineistonkeruu menetelmänä

Havainnoinnissa tutkija seuraa kokonaisvaltaisesti käyttäjää sen omassa toimintaympäristössään. Havainnoinnissa voidaan hyödyntää sanelukonetta tai videokameraa kynän ja paperin sijasta, varsinkin jos havainnoitava toiminta on monimutkaista ja vaikeasti ymmärrettävää. Toiminnan seuraamisella voi olla vaikutus myös itse toimintaan, jolloin tutkittavan toimintatavat muuttuvat ”oikeiksi”. (Sinkkonen ym. 2009, 100-103.)

Havainnoinnin avulla voidaan kerätä tietoa siitä mitä todella tapahtuu. Ihmisten käyttäytymistä ja arvostuksia tutkittaessa voidaan saada hyvin erilaista tietoa seuraamalla heitä arkielämässä, kuin mitä saataisiin kyselyillä. (Hirsjärvi ym. 2007, 207.) Passiivisessa havainnoinnissa voi tutkija yksinkertaisimmillaan seurata tutkittavaa ”taustalta” vaikuttamatta itse häneen. Tutkija voi myös tässä havainnointimuodossa kysellä tutkittavalta kysymyksiä, jos hän ei ymmärrä esimerkiksi tutkittavan motiiveja taikka toimintatapoja. (Sinkkonen ym. 2009, 101.)

6 Tietokantapohjainen verkkopalvelu

Luvussa kerrotaan lyhyesti kielistä, joita käytettiin projektissa tietokantapohjaisen verkkopalvelun tekemisessä. Kieliin lukeutuu HTML-merkintäkieli ja CSS-tyylikieli, joilla verkkosivuston runko toteutettiin. Varsinaisen sähköisen varausjärjestelmän tekemiseen käytettiin PHP-ohjelmointikieltä sekä SQL-kyselykieltä.

Luvussa kerrotaan myös tärkeimmistä projektissa käytetyistä ohjelmistoista. XAMPP-palvelinpaketti mahdollistaa sähköisen varausjärjestelmän kehittämisen ja testaamisen paikallisesti ilman tarvetta vuokrata ulkoista palvelintilaa. MariaDB on XAMPP-palvelinpaketin sisältämä relaatiotietokantajärjestelmä, jota käytettiin tietokantojen luomiseen ja hallitsemiseen. Adobe Dreamweaver on verkkosivujen luomiseen tarkoitettu työkalu, joka helpottaa verkkosivujen toteutusta.

6.1 HTML ja CSS

HTML lyhenne tulee sanoista HyperText Markup Language, ja sitä käytetään yleisimmin verkkosivujen rakentamiseen. HTML-dokumentit sisältävät pääosin vain tekstiä, ja esimerkiksi kuvien lisääminen HTML:ssä tehdään viittaamalla kuvatiedoston osoitteeseen HTML-dokumentissa. (Korpela 2005, 70.) HTML-kielen kehitystä valvoo World Wide Web Consortium (Larsen 2013, 2).

HTML-dokumentissa määritellään verkkosivujen rakenne, kuten mikä osa on otsikkoa, mitkä ovat kappaleita merkintöjä käyttäen. Esimerkiksi otsikko tason 1 tyylin tekstiä kirjoitetaan <h1> </h1> merkintöjen väliin, ja tämä kokonaisuudessaan muodostaa elementin. Vaikka HTML-kielellä määritetään verkkosivun rakenne, rakenteen tyyli määritetään käyttäen CSS-tyylikieltä. (Larsen 2013, 3-4).

CSS lyhenne tulee sanoista Cascading Style Sheets, ja CSS-kieltä käytetään luomaan sääntöjä verkkosivujen esittämiseen. CSS-kieltä voidaan käyttää HTML-dokumentin sisällä <style> elementillä, taikka linkittämällä erillinen CSS-tiedosto href-ominaisuudella HTML-dokumenttiin. CSS-kielen avulla voidaan määrittää sääntöjä jotka pätevät kaikkiin samanlaisiin elementteihin, kuten esimerkiksi <h1> elementteihin. (Larsen 2013, 191-193).

6.2 SQL

SQL eli Structured Query Language on relaatiotietokantahallintajärjestelmien ohjelmointikieli. SQL-kielellä voidaan käsitellä ja määritellä tietokantojen suhde rakenteita, kuten määrittelykokonaisuuksia ja taulukoita. SQL-kielellä ohjeistetaan hallintajärjestelmää mitä tietokannalle tehdään ja miten. SQL-ohjelmointikieli eroaa muista perinteisistä ohjelmointikielistä, sillä sitä käyttämällä ei ole mahdollista toteuttaa erillisiä tietokoneohjelmia, vaan sitä voidaan käyttää vain relaatiotietokantojenhallintajärjestelmissä. (Kriegel 2011, 9-10.)

SQL kieli on ISO ja ANSI standardoitu kieli. Standardit ovat merkityksellisiä SQL-kielelle, koska avoimeen lähdekoodiin perustuvien tietokantaratkaisujen, kuten MySQL ja PostgreSQL käyttö on yleistynyt. Uusin julkaistu standardi on SQL:2008. (Kriegel 2011, 10.)

6.3 PHP

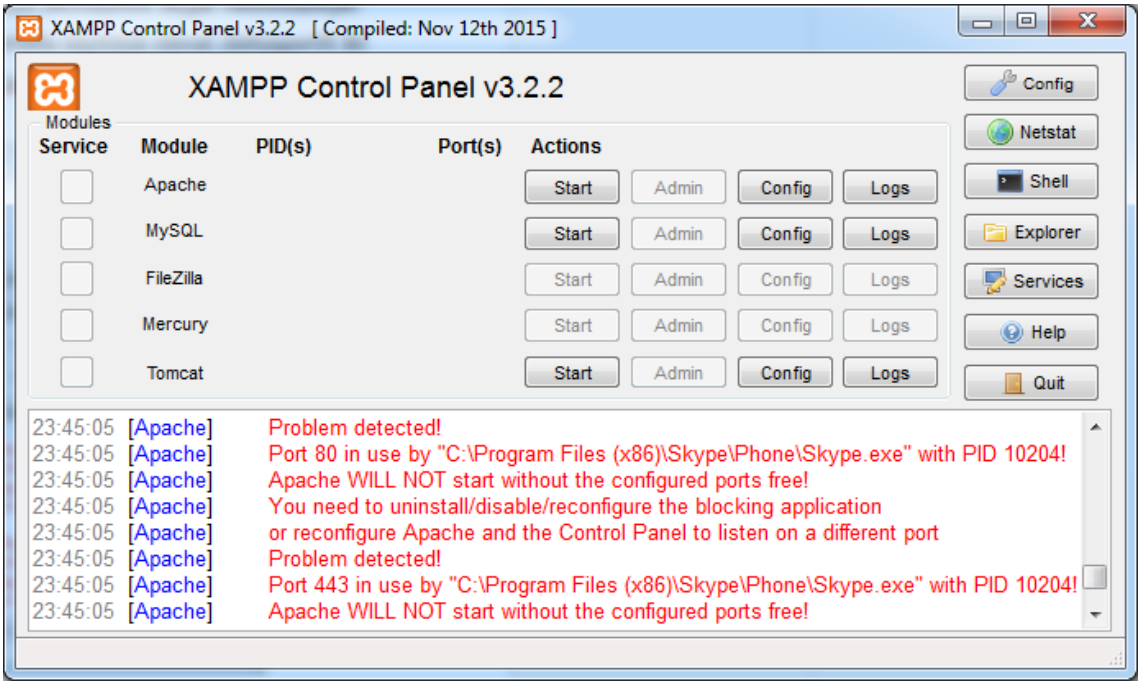
PHP on ohjelmointikieli joka juontaa juurensa vuodelle 1995, jolloin PHP lyhenne tuli sanoista Personal Home Page. Silloin PHP-ohjelmakoodi sisälsi kaksi toimintoa, verkkosivujen vierailijoiden tietojen kirjaamisen ja vierailijoiden määrän näyttämisen. PHP:stä tuli nopeasti hyvin suosittu, mikä on johtanut uusien versioiden kehittämiseen. (Gilmore 2008, 2.) PHP version 3 myötä PHP lyhyenne on siirtynyt tarkoittamaan Hypertext Preprocessor (Rantala 2002, 12).

PHP on tulkattava ohjelmointikieli, jota suoritetaan joka kerta kun palvelin lähettää verkkosivun selaimelle. PHP-ohjelmointikieli on oliopohjainen, sekä se sisältää tyypilliset ohjelmointikielten rakenteet kuten silmukat, ehtolauseet, muuttujat sekä funktiot. PHP-ohjelmointikieltä voidaan käyttää kirjoittamalla sitä suoraan HTML-ohjelmakoodin sisälle, taikka kirjoittamalla sitä erilliseen tekstitiedostoon. (Heinisuo & Rauta 2007, 13.)

6.4 XAMPP kehitysympäristö

XAMPP on avoimen lähdekoodin palvelinpaketti, jonka lyhenne tulee sanoista Cross-platform (X), Apache, MariaDB, PHP ja Perl. XAMPP-palvelinpaketti on järjestelmäriippumaton ja siitä löytyy toimivat versiot Windows-, Linux-, sekä OS X-käyttöjärjestelmille. (Apache friends 2016.) Palvelinpaketti sisältää kaiken tarvittavan palvelimen pystyttämiseen.

Projektia varten ladattiin palvelinpaketti XAMPP. XAMPP mahdollistaa paikallisen kehitysympäristön käyttöönoton, jonka avulla voitiin rakentaa ja testata verkkosivustoa, ennen sen lataamista toimeksiantajan palvelimelle. XAMPP:in käyttöönotto on yksinkertainen prosessi, jossa peruskäyttäjän ei tarvitse kuin ladata ilmainen asennuspaketti sen kotisivuilta, sekä asennuksen aikana painella vain seuraava-painiketta. Isoin ongelma XAMPP:in käyttöönottamisessa on Apache:n käyttämät oletusportit, jotka sattuvat olemaan samat kuin Microsoftin Skype-viestintäohjelmistossa. Ongelma on kuitenkin helppo ratkaista vaihtamalla käytössä olevat oletusportit 80 ja 443 Apachen asetustiedostoista vapaina oleviin portteihin. Kuviossa 4 näkyy XAMPP:in ohjauspaneelissa virheilmoitus käytössä olevista porteista.



Kuvio 4: Virheilmoitus XAMPP:in ohjauspaneelissa

6.5 MariaDB

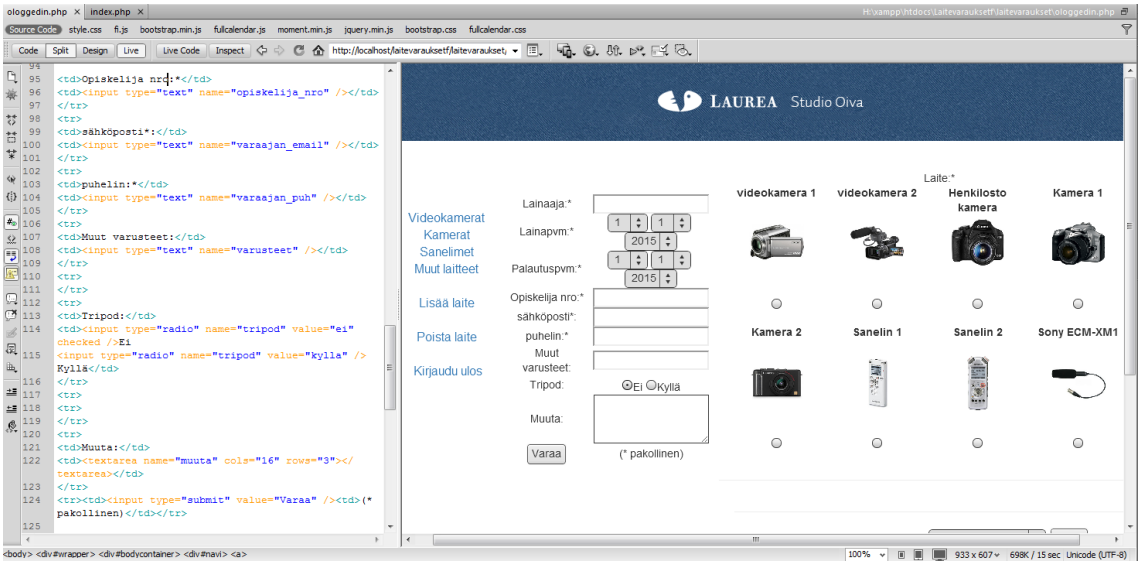
MariaDB on avoimen lähdekoodin relaatiotietokantahallintajärjestelmä, jonka tekivät MySQL:n alkuperäiset kehittäjät. MariaDB on tehty parannetuksi korvikkeeksi MySQL:le, ja se tarjoaa SQL-käyttöliittymän tiedon käsittelyyn. Merkittäviä MariaDB käyttäjiä ovat Wikipedia, Facebook ja Google. (MariaDB 2016.) Kuviossa 5 on MariaDB:n logo.

Kuvio 5: MariaDB:n logo

Paikallisessa kehitysympäristössä käytettiin XAMPP-palvelinpaketin sisältämää MariaDB-relaatiotietokantahallintajärjestelmää. Valinta MariaDB:n käyttämiseen johtui sen tietokantojen yhteensopivuudesta MySQL-tietokantojen kanssa. Tietokantojen yhteensopivuus mahdollistaa paikallisessa kehitysympäristössä luotujen tietokantojen käyttämisen toimeksiantajan MySQL-palvelimella.

6.6 Adobe Dreamweaver

Dreamweaver on Adobe Systemsin maksullinen ohjelmisto, joka on tarkoitettu verkkosivujen kehitykseen. Dreamweaver sisältää useita hyödyllisiä toimintoja, kuten koodin täydennystä, väritystä, ja jäsennystä. Ohjelmisto sisältää myös toiminnon, jonka avulla käyttäjä pystyy tarkastelemaan verkkosivuston yhteensopivuutta eri selainten kanssa, ja yhteensopivuusongelmien ilmentyessä ohjelmisto kertoo niitten syyt ja seuraukset. Kuviossa 6 näkyvä Dreamweaverin split-näkymä mahdollistaa ohjelmakoodin vaikutuksen seuraamista sivuston ulkonäköön.



Kuvio 6: Dreamweaverin split-näkymä

Projektin verkkosivujen toteutukseen valittiin Adobe Dreamweaver -ohjelmisto, koska se löytyi Studio Oivan työpisteeltä valmiiksi asennettuna. Dreamweaver tukee kaikkia ohjelmointikieliä, joita tiedettiin tarvittavan projektissa. Minulta löytyi myös aikaisempaa kokemusta Dreamweaverin käytöstä, joten erillistä perehdyttämistä sen käyttämiseen ei vaadittu.

7 Projektin toteutus ja kulku

Tässä luvussa käydään läpi opinnäytetyöhön kuuluvan projektin kulkua ja sen eri vaiheita. Projekti alkoi vaatimusmäärittelyllä, jonka jälkeen pohdittiin toimeksiantajalta löytyvän vanhan varausjärjestelmän hyödyntämistä uudessa järjestelmässä. Päädyttiin tulokseen, että kannattaa käyttää vanhan järjestelmän osia hyväksi, jos niitä pidetään hyvinä ja toimivina.

Aloin havainnoida vanhan varausjärjestelmän käyttäjiä heti projektin alettua, tarkoituksena saada tietoa, jonka pohjalta uutta järjestelmää voidaan suunnitella. Seuraava vaihe projektissa oli toimintatarinan ja käyttötarinan tekeminen, ja näitten avulla järjestelmän prototypointi. Hyväksi todetun prototyypin jälkeen oli aika alkaa toteuttaa varsinaista järjestelmää

HTML-, CSS-, PHP-, ja MySQL-kielillä. Projektin viimeisiä vaiheita olivat sivuston testaus, sivujen julkaiseminen toimeksiantajan verkkosivuilla, sekä sivuston ylläpito.

Sähköisen varausjärjestelmän tekemisessä on vaihejakomallina sovellettu vesiputousmallin ja ketterien menetelmien yhdistelmää. Projektin kulku oli suurimmaksi osaksi lineaarinen, mutta iterointia tehtiin, kun se koettiin tarpeelliseksi. Varausjärjestelmän osia pyrittiin näyttämään toimeksiantajalle niitten valmistuttua ketterien menetelmien mukaisesti. Varausjärjestelmä otettiin myös käyttöön ennen kuin sen kaikki ominaisuudet olivat valmiita.

7.1 Vaatimusmäärittely

Sähköisen varausjärjestelmän toteutus alkoi vaatimusmäärittely tapaamisella toimeksiantajan kanssa. Vaatimusmäärittelyssä käytiin läpi projektin tavoitteita ja vaatimuksia yleisesti, jättäen minulle suhteellisen vapaat kädet projektin toteuttamiseen. Projektin tavoitteisiin kuuluivat kalenteritoiminto, sekä erilliset käyttäjätunnukset oppilaille, Studio Oivan henkilöstölle, ja Laurean henkilöstölle. Erillisillä käyttäjätunnuksilla voidaan rajata pois opiskelijoitten pääsy osaan järjestelmän toiminnallisuuksia, sekä käyttäjäoikeuksia hyödyntäen järjestelmässä olisi mahdollisuus pitää laitteita joihin esimerkiksi opiskelijoilla ei ole lainausoikeutta. Projektin vaatimuksia olivat helpot tavat lisätä ja poistaa laitteita järjestelmästä, sekä tapa seurata varauksia. Sähköisen varausjärjestelmän kohderyhmiä ovat Laurean opiskelijat, henkilöstö ja Studio Oivan henkilöstö.

Vaatimusmäärittelyssä toimeksiantaja kertoi myös heidän vanhasta sähköisestä varausjärjestelmästä. Vanha varausjärjestelmä oli kuitenkin tuntemattomasta syystä toimimaton. Vaatimusmäärittelyssä sovittiin, että projektissa voidaan jatkokehittää vanhaa järjestelmää taikka käyttää sen osia hyväksi, jos projektin kehittäjä näkee tämän parhaaksi vaihtoehdoksi.

7.2 Vanha järjestelmä

Toimeksiantajan vanhan sähköisen varausjärjestelmän tutkiminen alkoi selvittämällä syytä sen toimimattomuuteen, ja syyksi paljastuikin tietokantayhteydet, jotka pystyttiin korjaamaan nopeasti. Vanha järjestelmä oli toteutettu yhdelle käyttäjäryhmälle, Studio Oivan Henkilöstölle. Järjestelmän visuaalinen ilme jätti paljon toivomisen varaan, siitä puuttui esimerkiksi selkeä visuaalinen hierarkia, jonka takia järjestelmän ulkoasu oli epäselvä. Laitteiden poistamistoiminto puuttui myös kokonaan.

Vanhan järjestelmän ohjelmakoodia tutkiessa kävi selväksi, että sen ymmärtäminen ja muokkaaminen olisi aikaa vievä prosessi. Toimeksiantajan kanssa sovittiin, että uuden varausjär-

jestelmän toteuttaminen on parempi vaihtoehto, kuin aikaisemman järjestelmän jatkokehitys. Osia vanhasta järjestelmästä olisi kuitenkin mahdollista käyttää hyväksi uuden varausjärjestelmän toteuttamisessa.

7.3 Havainnointi käyttäjätutkimuksena

Ollessani työharjoittelussa Studio Oivassa suoritin käyttäjätutkimuksen muotona passiivista havainnointia, jossa seurasin opiskelijoiden kanssakäymistä Studio Oivan kanssa. Havainnoinnissa toimin palveluhenkilönä, joka vastasi Studio Oivan lainauspalvelusta paperisella varausjärjestelmällä. Havainnoinnin tarkoituksena oli saada tietoa opiskelijoiden tarpeista, käyttäytymisestä ja toimintatavoista, jolla saavutettaisiin parempi pohja sähköisen varausjärjestelmän suunnitteluun.

Havainnoinnin avulla huomasin, että opiskelijat saattoivat unohtaa lainattujen laitteiden palauttamisen, tai varattujen laitteiden hakemisen ajallaan. Sähköisen varausjärjestelmän varauslomakkeen tulisikin sisältää puhelinnumero- ja sähköposti-kentät jotka helpottaisivat yhteydenpitoa opiskelijoihin. Varausjärjestelmän tulisi myös sisältää toiminnot, joilla seurata onko laitteita haettu lainaan, taikka palautettu lainasta. Opiskelijat saattoivat myös peruuttaa heidän lainavarauksia koska he saivat tarvittavan lainan kätevämmin muualta. Varausjärjestelmässä tulisikin tämän takia olla toiminto jolla poistaa varauksia. Opiskelijoille kaikkien laitteiden käyttö ei myöskään ollut entuudestaan tuttua, jolloin he tarvitsivat opastusta näitten käytössä. Sähköiseen varausjärjestelmään olisikin ehkä tarpeellista lisätä laitesivut, johon voidaan muun muassa lisätä tietoja laitteista, ja linkit laitteiden ohjekirjoihin.

7.4 Varauksen tekemisen toimintatarina ja käyttötarina

Tämä alaluku sisältää toimintatarinan, jonka tarkoitus on auttaa hahmottamaan paperisen varausjärjestelmän puutteita ja ongelmia. Huomaamalla puutteet ja ongelmat, on mahdollista korjata ne uudessa järjestelmässä. Alaluku sisältää myös käyttötarinan, jonka tarkoitus on testata kyseistä suunnitelmaa.

Tarinoita varten rakennettiin persoona palvelua käyttäneen opiskelijan pohjalta, joka näkyy taulukossa 1. Toimintatarina on saatu palvelua käyttäneeltä opiskelijalta suoraan. Käyttötarina on rakennettu toimintatarinan pohjalta.

Nimi:	Olli Opiskelija
Asuinpaikka:	Vantaa
Ikä:	26
Koulutus:	Datanomi
Harrastukset:	Lenkkeily, frisbeegolf ja lukeminen
Verkkoyhteys:	100/10Mbit/s
Verkon käyttö:	Sähköposti, pelailu, koulutehtävien hakeminen koulun intrasta

Taulukko 1: Olli Opiskelija persoona

7.4.1 Toimintatarina

Olli Opiskelija sai koulussa ruotsintunnilla ryhmätyöksi tehtävän, jossa heidän piti kuvata itsestään ruotsinkielinen yritysesittely heidän valitsemastaan yrityksestä. Tehtävää varten opettaja kertoi ryhmille, että koulussa toimivalta Studio Oivalta on mahdollisuus lainata videokamera tehtävää varten. Ryhmässä sovittiin alustavaksi kuvausaikatauluksi seuraava viikko, ja että Olli saa hoitaa videokameran lainauksen Studio Oivasta.

Kotia päästyään Olli meni tietokoneella tutkimaan Studio Oivan verkkosivuja. Verkkosivuilta Olli löysi Studio Oivan yhteystiedot ja tiedon että videokameran voi varata käymällä heidän toimistolla. Olli menikin heti seuraavana aamuna käymään Studio Oivassa, jossa hänelle kerrottiin, että valitettavasti kaikki videokamerat ovat varattuja seuraavan viikon, mutta sitä seuraavalla viikolla videokameroita olisi vapaana. Olli soitti ryhmäläisilleen ja tiedusteli, että kävisikö heille kuvausaikatauluksi viikko, jolloin videokameroita on vapaana. Ollin ehdottama kuvausaikataulu sopi ryhmäläisille, ja Olli varasikin videokameran ollessaan vielä paikan päällä.

Toimintatarina auttoi huomaamaan paperisen varausjärjestelmän heikkouksia sekä sähköisen varausjärjestelmän tarpeellisuuden. Kursseilla useat oppilaat saavat saman tehtävän, jolloin he tarvitsevat myös samaa laitetta tehtävää varten. Oppilaitten on hyvä nähdä verkossa olevasta varausjärjestelmästä laitteiden varaustilanne, vaikka kalenteritoiminnon avulla.

7.4.2 Käyttötarina

Olli Opiskelija sai videotuotannon kurssilla tehtäväksi kuvata ja editoida videon hänen valitsemastaan yrityksestään. Kurssin opettaja kertoi mahdollisuudesta lainata videokameroita koulussa toimivasta Studio Oivasta.

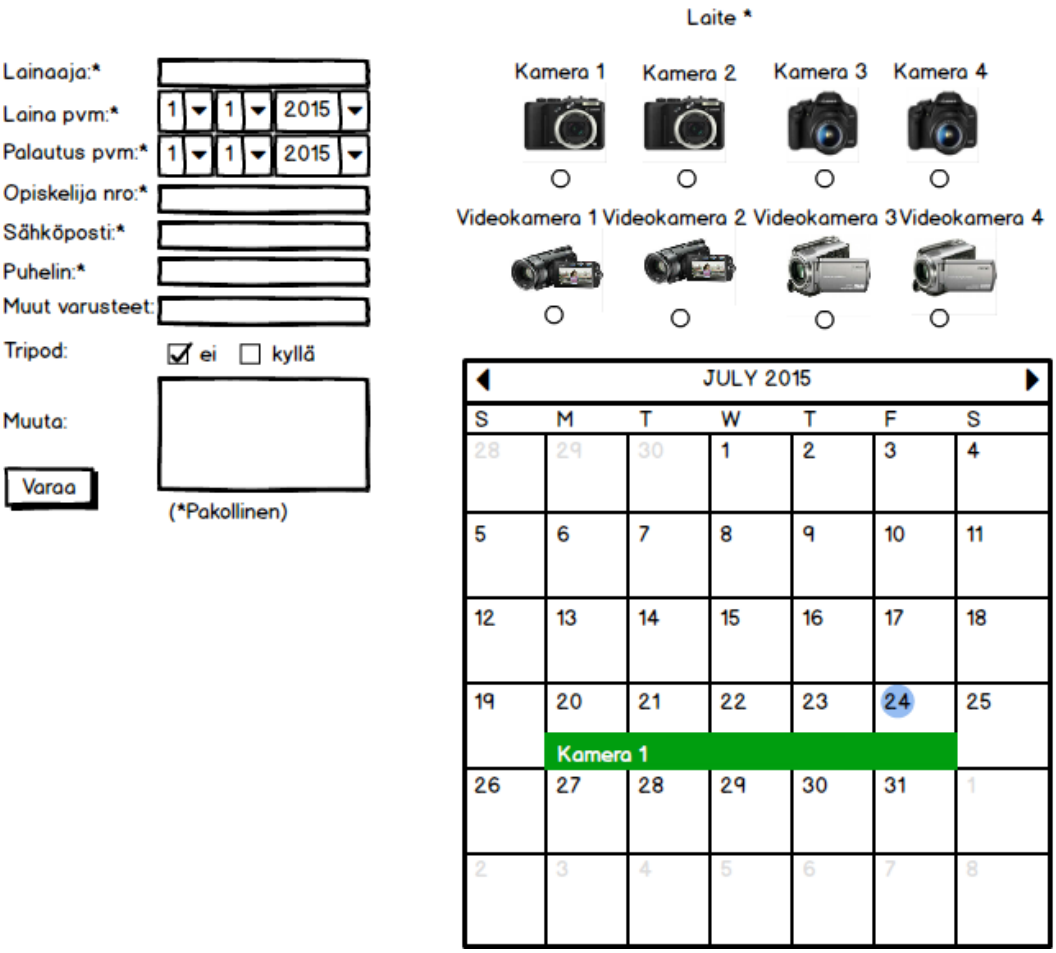
Olli avasi verkkoselaimen ja avasi Studio Oivan verkkosivut. Verkkosivuilta Olli löysi linkin laitevaraukseen, sekä opiskelijatunnukset sinne. Olli painoi laitevaraus-linkkiä, joka ohjasi hänet varausjärjestelmän kirjautumissivulle. Olli kirjautui järjestelmään oppilastunnuksilla, jonka jälkeen varausjärjestelmä ohjasi Ollin varaussivulle. Varaussivulla Olli huomasi listan laitteita ja kalenterin, josta näkyi varattuna olevat laitteet. Olli löysi nopeasti sopivan vapaana olevan videokameran, joka sopi hänen tehtäväänsä. Olli valitsi videokameran ja täydensi varauslomakkeen yhteystiedoillaan, varauksen alku- ja päätöspäivällä alasvetovalikosta, sekä opiskelijanumerollaan. Olli painoi vielä lopuksi varaa-painiketta, jonka jälkeen varaus oli tehty ja kalenteriin ilmestyikin hänen juuri tekemänsä varaus. Olli kirjautui järjestelmästä pois kirjaudu ulos-painikkeella, joka vei hänet kirjautumissivulle.

Käyttötarina antoi mahdollisuuden kokeilla suunnitelmaa järjestelmästä, ja se antoi hyvän pohjan prototypoinnin aloittamiselle. Käyttötarinasta heräsi kysymys opiskelijatunnuksien järkevimmästä jakelutavasta. Tarinassa kaikilla opiskelijoilla on yhdet yhteiset tunnukset järjestelmään, ja ne näkyvät kaikille Studio Oivan sivuilla. Tunnuksia olisi mahdollista väärinkäyttää esimerkiksi turhien varauksien luomiseen.

7.5 Prototyyppi

Prototyypeillä testataan osia järjestelmästä, kuten esimerkiksi käyttöliittymää. Prototyyppejä käytetään iteratiivisessa kehittämisessä, jossa niitä kehitetään sykleissä. Yhteen sykliin kuuluu olemassa olevan suunnitelman analysointi, täydentäminen, sekä prototyypin tekeminen ja sen testaaminen. Prototyyppiä iteroidaan eteenpäin niin kauan, kunnes päästään prototyypin hyvään käytettävyyteen ja toiminnallisuuteen. Prototyypit ovat hyvä työkalu suunnittelussa koska ne mahdollistavat palvelun testaamisen jo ennen varsinaista ohjelmointia. Prototyyppien tekeminen kannattaa aloittaa yksinkertaisista paperiprototyypeistä, koska niitä on helppoa ja nopeata luonnostella paperille. Käyttäjien arvostelukynnys on myös alhaisempi, kun tuote on vasta käsin luonnosteltu. (Sinkkonen ym. 2009, 204-205.)

Prototyyppien tekeminen aloitettiin paperiprototyypeistä, jotka luonnosteltiin paperille. Paperiprototyyppejä tekemällä pystyttiin suorittamaan iterointikierroksia nopeasti, ja niitä tehtiin niin kauan, kunnes saavuttiin prototyyppiin jonka toiminnallisuus ja käytettävyys todettiin hyväksi. Kuvion 7 prototyyppi, on viimeinen varaustoiminnon prototyyppi, ja sitä testattiin käyttäjän avulla. Käyttäjän tehtävä oli testissä luoda laitevaraus, joka onnistui ongelmitta.

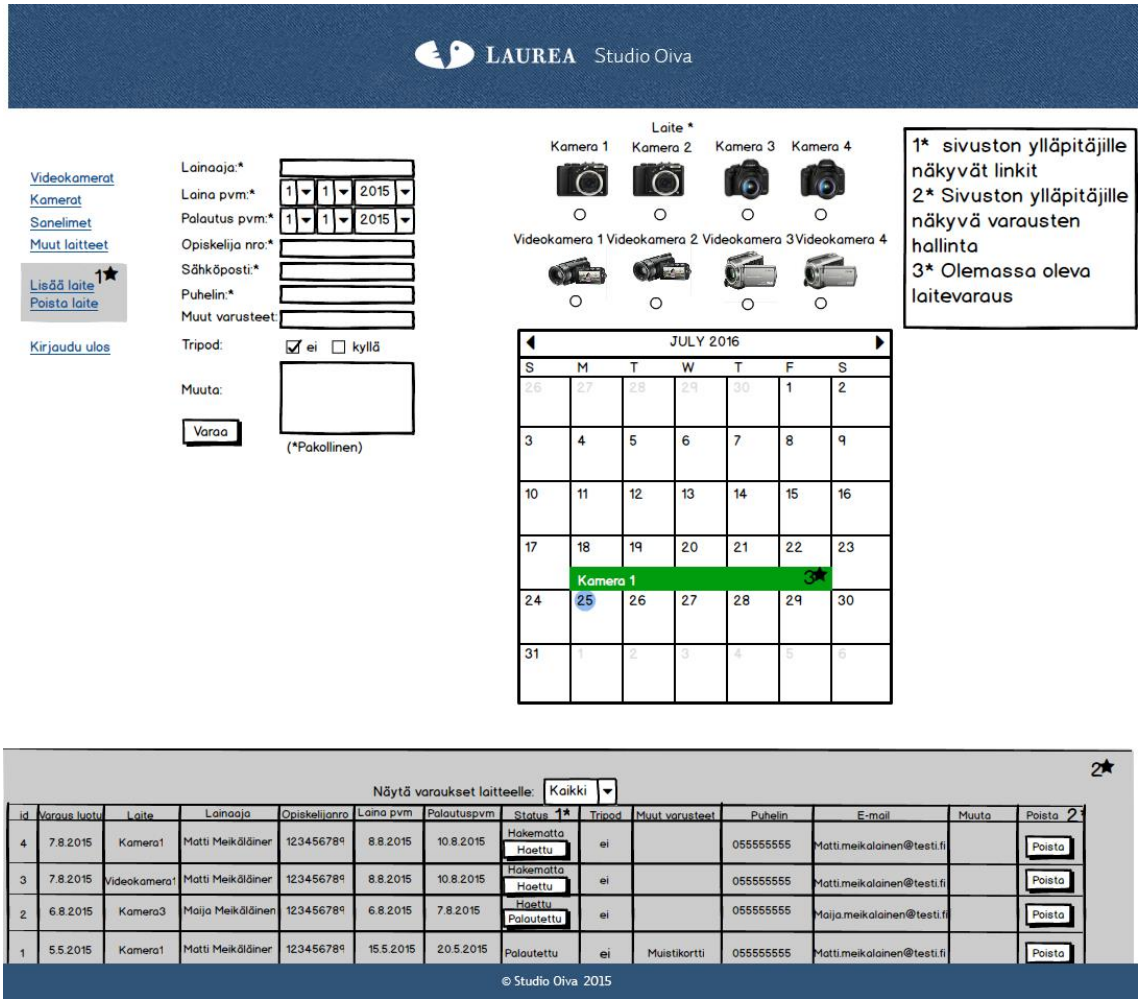


Kuvio 7: Varaustoiminnon prototyyppi

7.6 Rautalankamalli

Rautalankamallilla kuvataan sivuston rakennetta, ja visuaalista hierarkiaa. Se voi olla yksinkertainen kuva joka on toteutettu kuvankäsittelyohjelmalla, taikka yksityiskohtainen prototyyppi joka mahdollistaa sivun elementtien ja toimintojen yksityiskohtaisen dokumentoinnin. Dynaamisen sivun rautalankamallin tulisi esitellä kaikki sivuilla tapahtuvat muutokset, sekä kertoa milloin muutokset tapahtuvat. (Sinkkonen ym. 2009, 212-213.)

Kuvion 8 rautalankamalli toimi varausjärjestelmän pohjapiirustuksena, jonka perusteella varsinaiset sivut toteutettiin. Rautalankamallissa näkyy varaussivun rakenne ja visuaalinen hierarkia. Varausjärjestelmän osat, mitkä ovat vain järjestelmävalvojien käytössä, on merkitty harmailla laatikoilla kuviossa.

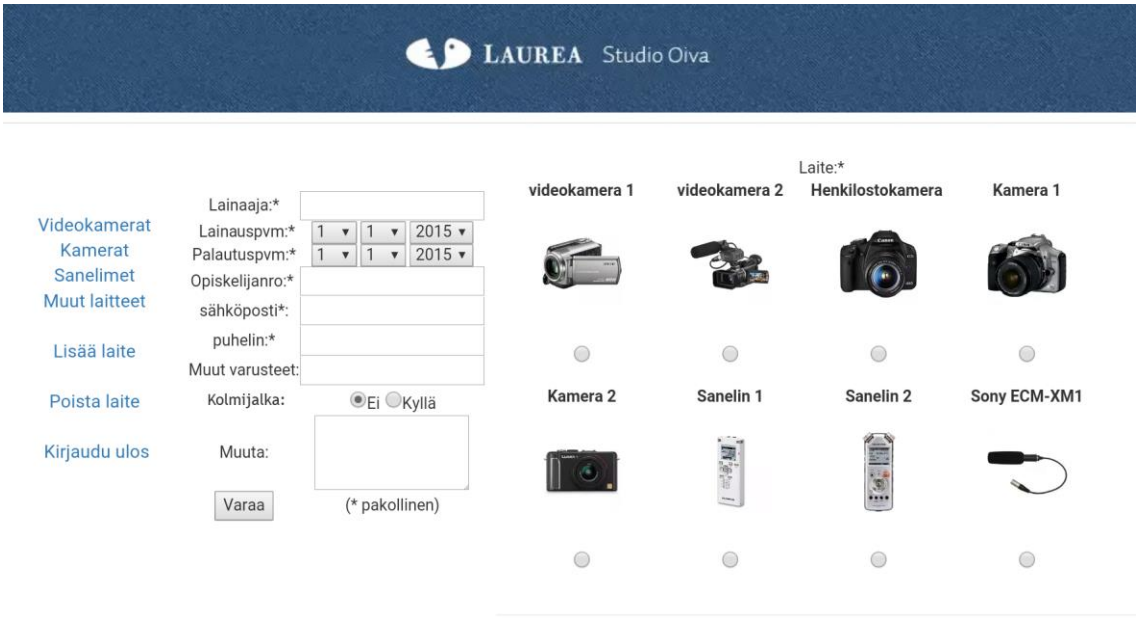


Kuvio 8: Varaussivun rautalankamalli

7.7 Varausjärjestelmän testaus ja julkaisu

Verkkosivuston toimivuus testattiin suosituimmilla selaimilla, jotka testauksen aikana olivat W3Schoolsin mukaan Chrome, Firefox sekä Internet Explorer. Verkkosivustoa testattiin myös Windows 7-, Windows 10-, ja Ubuntu 12.04.5-käyttöjärjestelmillä, sekä Android-käyttöjärjestelmää käyttävällä Nexus 5-puhelimella. Selainten toimivuus testattiin käyttämällä varausjärjestelmää kyseisillä selaimilla, sekä yhteensopivuuden takaamiseksi käytettiin myös Adobe Dreamweaverin selainten yhteensopivuuden tarkastustoimintoa.

Varausjärjestelmä suunniteltiin pöytäkoneitten käyttäjiä ajatellen. Sivustosta ei tämän takia tehty erillistä mobiiliversiota mobiilikäyttäjiä varten, eikä sivustoa myöskään ole erityisesti optimoitu mobiilille. Sivusto toimii kuitenkin mobiililaitteilla ongelmitta, mutta esimerkiksi varauslomakkeen kentät voisivat olla suurempia mobiilikäyttäjillä. Kuviossa 9 on mobiililaitteella otettu kuvakaappaus varaustoiminnosta.



Kuvio 9: Varaustoiminto Nexus 5 puhelimella

Sivuston toimivuuden varmistuttua testauksen perusteella, todettiin sivuston olevan julkaisuvalmis. Julkaisu tapahtui lataamalla sähköisen varausjärjestelmän sivut toimeksiantajan käyttämälle palvelimelle WinSCP tiedostonsiirto-ohjelmalla. Varausjärjestelmän tietokannat kopioitiin myös toimeksiantajan MySQL-palvelimelle, jonka jälkeen sivusto oli julkaistu ja toiminnassa.

7.8 Sivujen ylläpitovaihe

Järjestelmän julkaisun jälkeen olin vielä työharjoittelussa toimeksiantajalla noin puolivuotta. Tänä aikana pystyin ylläpitämään, jatkokehittämään, sekä korjaamaan sivustosta löytyviä ongelmia. Havainnoin myös uuden järjestelmän vaikutuksia Studio Oivan lainauspalveluun tänä aikana.

Sivusto julkaistiin alun perin ilman kalenteritoimintoa ja pelkästään Studio Oivan henkilöstön käyttöön. Tarkoitus kuitenkin oli lisätä kalenteritoiminto, sekä ottaa opiskelijatunnukset ja henkilöstötunnukset käyttöön niin nopealla aikataululla kuin mahdollista. Kalenteritoiminto saatiin lisättyä järjestelmään pari viikkoa sivuston julkaisemisen jälkeen, jonka jälkeen myös opiskelijatunnukset otettiin käyttöön.

Järjestelmän opiskelijatunnukset olivat käytössä muutaman viikon, ja tunnukset löytyivät tällöin varausjärjestelmän kirjautumissivulta. Toimeksiantajan pyynnöstä opiskelijatunnukset otettiin kuitenkin etusivulta pois. Henkilöstötunnukset piti alun perin ottaa käyttöön onnistuneen kokeilun opiskelijatunnuksien kanssa, mutta kummatkin käyttäjäryhmät jäivät loppujen lopuksi hyödyntämättä.

Sähköinen varausjärjestelmä yksinkertaisti ja helpotti varausprosessia huomattavasti, vaikkakin järjestelmää käyttivät vain Studio Oivan henkilöstö. Käyttöönoton jälkeen varauksia pystyttiin ottamaan vastaan sähköposteilla ja puhelinsoitoilla muualtakin kuin Studio Oivan toimistolta. Verrattuna vanhaan varausjärjestelmään, uutta järjestelmää oli myös helpompi käyttää useamman henkilön toimesta samanaikaisesti.

8 Järjestelmän rakenne ja toiminnot

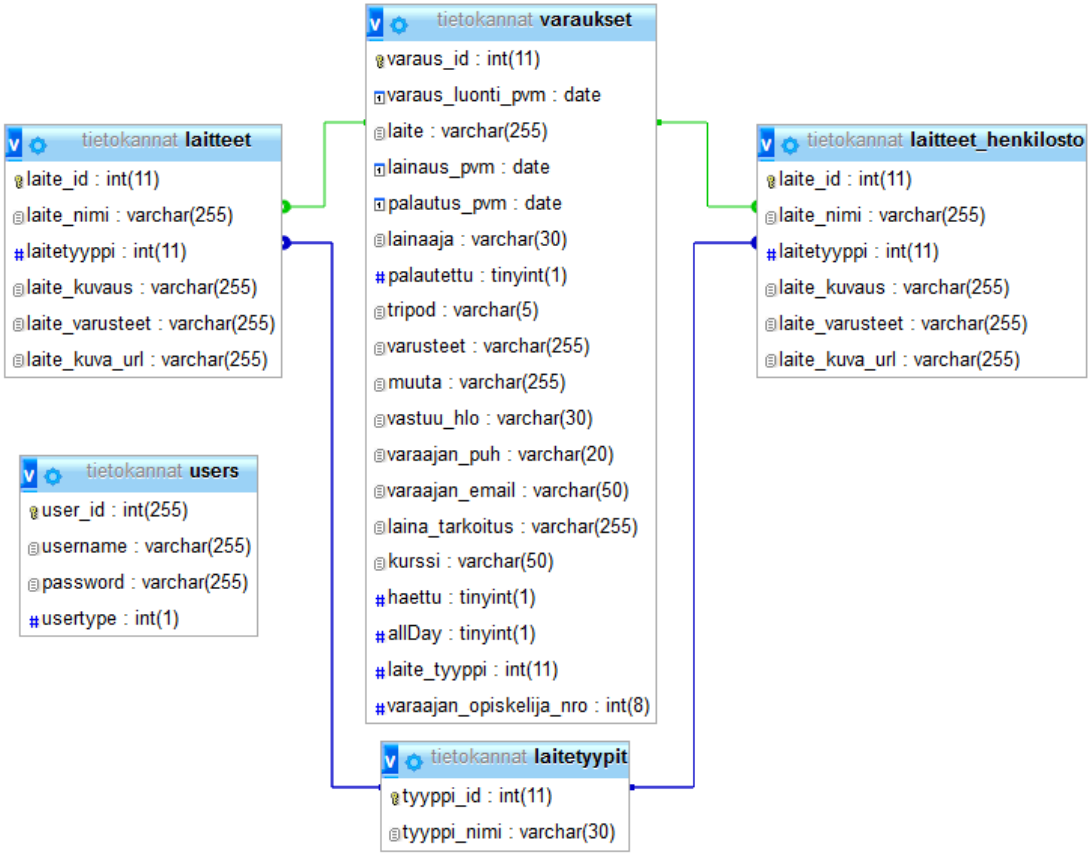
Luvussa tutustutaan järjestelmän tietokantoihin, jotka luotiin MariaDB-relaatiotietokantahallintajärjestelmällä. Tietokantojen käsittelyyn käytetään PHP:n MySQLi-lisäosaa, vanhan MySQL-lisäosan sijasta. Valinta MySQLi- ja MySQL-lisäosien välillä oli selvä, koska vanha lisäosa ei tue MySQL:n 4.1.3 ja uudempien versioiden toimintoja, sekä se on poistumassa PHP 7 -version myötä käytöstä.

Luvussa käydään läpi uuden varausjärjestelmän verkkosivuston rakennetta, sekä sivurunkoa joka on sama jokaisella järjestelmän sivulla. Luvussa tutustutaan myös varausjärjestelmän tärkeimpiin PHP:llä toteutettuihin toimintoihin, joihin kuuluvat varaustoiminto ja varauksien hallinta. Varausjärjestelmän toteutukseen käytettiin PHP 5.6.8 -versiota, joka tuli projektin aikana uusimman XAMPP-palvelinpaketin mukana.

8.1 Tietokannat

Vanhan varausjärjestelmän tietokantarakenne oli toimiva, joten sitä päätettiin käyttää sitä hyväksi. Tietokantarakenne pysyi pääpirteittäin samana, muutamia muutoksia lukuun ottamatta. Henkilöstölle lainattaville laitteille syntyi uusi laitteet_henkilosto-taulukko, sekä käyttäjät-ja varaukset-taulukkoihin lisättiin muutama uusi sarake.

Järjestelmän tietokantarakenne on kuvion 10 mukainen. Users-taulukko sisältää käyttäjätunnukset, salasanat sekä käyttäjätyypit, joita järjestelmässä on opiskelijat, henkilöstö ja järjestelmänvalvoja. Varaukset-taulukko sisältää tiedot varauksista, kuten varaajan, laitteen, käyttötarkoituksen ja yhteystiedot. Laitetyypit-taulukko sisältää tiedon laitteiden tyypistä. Laitteet-taulukko sisältää tiedot peruslaitteista, sekä laitteet_henkilosto-taulukko sisältää tiedot erikoislaitteista, jotka eivät ole opiskelijoitten lainattavissa. Avaimen kuva merkitsee sarakkeen pääavaimeksi, ja viivat taulujen välillä kuvaavat taulukkojen välisiä suhteita.

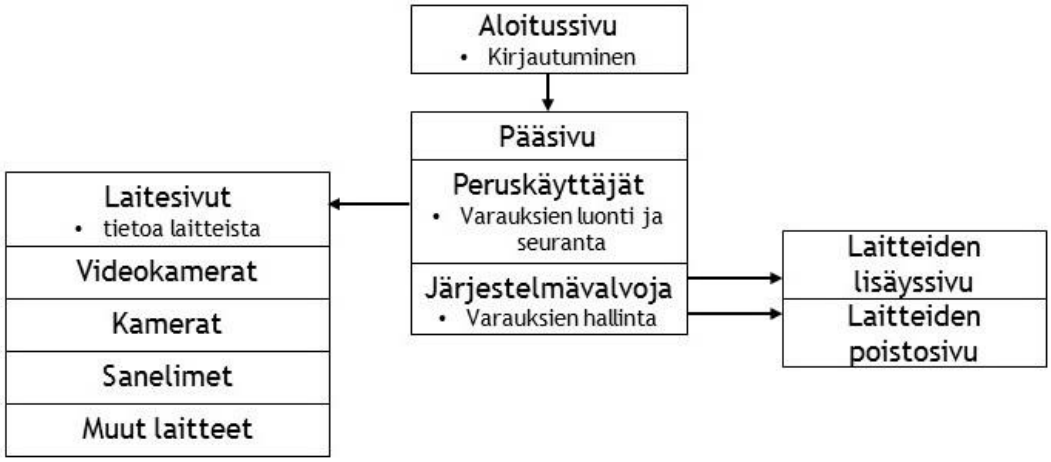


Kuvio 10: Tietokantarakenne

Pääavainten tarkoitus tietokannoissa on suojella tietokantojen eheyttä, sekä ne auttavat myös luomaan yhteyksiä taulukkojen välille. Pääavain on aina ainutlaatuinen tunniste riville tietoa taulukossa, ja pääavain sarake ei voi olla tyhjä. Vierasavaimet taas usein ovat toisen taulukon pääavaimia, ja niitten ei tarvitse olla ainutlaatuisia taulukossa. Eheyden suojelu näitten avulla tapahtuu esimerkiksi estämällä taulukosta tiedon poistamista, jos niitten sisältämää tietoa on käytössä vierasavaimena (Kriegel 2011, 81-86). Varausjärjestelmän tietokannoissa esimerkiksi laitteet-taulukon laitetyyppi-sarake on vierasavain laitetyypit-taulukosta, ja laitetyypit-taulukosta ei voi poistaa laitetyyppiä, joka on käytössä laitteet-taulukossa.

8.2 Sivuston rakenne

Sähköiseen varausjärjestelmään on luotu kolme eri käyttäjäryhmää, opiskelijat ja henkilöstö, jotka ovat peruskäyttäjiä, sekä järjestelmänvalvojat. Järjestelmä eroaa opiskelijoitten ja henkilöstön välillä vain heille näytettävissä laitteissa. Kuvio 11 on kaavio sivuston rakenteesta.



Kuvio 11: Sivuston rakenne

Aloitussivu sisältää kirjautumisen, joka ohjaa käyttäjän pääsivulle onnistuneen kirjautumisen jälkeen. Peruskäyttäjälle pääsivulla näkyvät varauksen luonti-lomake, sekä kalenteri mistä seurata varauksien tilaa. Pääsivulta peruskäyttäjä pystyy myös kirjautumaan ulos, taikka siirtyä laitesivuille, joita löytyy jokaiselle laitetyypille oma. Laitesivut sisältävät kuvauksen laitteista, sekä tiedon mahdollisista lisävarustuksista.

Järjestelmänvalvoja käyttäjäryhmällä on pääsy kaikkiin toimintoihin jotka ovat peruskäyttäjillä, sekä työkalut hallita varausjärjestelmää. Pääsivu sisältää käyttäjäryhmälle varausten-hallintatyökalut joilla voi poistaa varauksia, sekä mahdollisuus hallita laitteiden tilaa merkitsemällä niitä haetuksi sekä palautetuksi. Järjestelmänvalvojalla on myös pääsy pääsivulta laitteiden lisäyssivulle sekä poistosivulle.

8.3 Sivurunko

Jokainen sivu sähköisessä varausjärjestelmässä koostuu samankaltaisesta sivurungosta, joka on tehty HTML- ja CSS-kieliä käyttäen. Jokainen sivu sisältää myös pienen määrän PHP-ohjelmointikieltä, mutta suurin osa PHP:llä tehdyistä toiminnoista on kirjoitettu omiin tiedostoihinsa. Varausjärjestelmässä PHP:ta käytetään varausjärjestelmän toimintojen luomiseen, tietokantayhteyden muodostamiseen, sekä tuomaan sivustoon dynaamisuutta. Suurin osa verkkosivuston käyttämistä PHP-toiminnoista on kirjoitettu omiin tiedostoihinsa. PHP-tiedostot sekä FullCalendar-liitännäinen ovat liitetty sivustoon käyttämällä vastaavaa ohjelmakoodia kuin kuviossa 12.

```
<?
 include("esimerkki.php")
?>
```

Kuvio 12: Include-komento

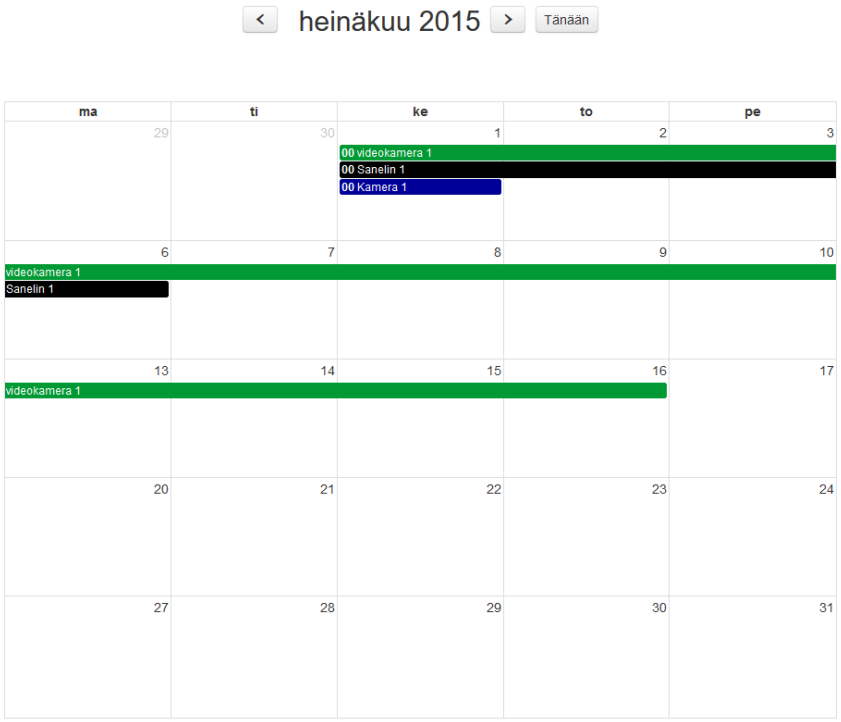
Jokainen sivu koostuu yläpalkin, keskiosan, sekä alapalkin div-elementeistä, joitten näyttämissäännöt ovat kirjoitettu erilliseen CSS-tyylitiedostoon, joka liitetään sivuille HTML:n href-ominaisuudella. Yläpalkki koostuu Studio Oivan logosta. Keskiosa sisältää pystysuuntaisen navigaatio-palstan, joka sisältää linkit jokaiselle erillisille sivulle sivustossa. Varausjärjestelmän toiminnot, kuten laitteiden varaus, varauksien hallinta, sekä kalenteritoiminto on myös sisällytetty sivuston keskiosaan. Sivuston alaosassa koostuu alatunnisteesta, jossa kerrotaan, että sivuston tekijänoikeudet kuuluvat Studio Oivalle. Kuviossa 13 on sisäänkirjautumisen div-elementin määrittely CSS-tyylitiedostossa.

```
#login {
    Margin: auto;
    Width: 250px;
    Height: 300px;
}
```

Kuvio 13: Login div-elementti

8.4 FullCalendar kalenteritoiminto

FullCalendar on avoimen lähdekoodin jQuery-liitännäinen, jonka avulla voidaan lisätä verkkosivuille tapahtumakalenteri. Kalenteri on hyvin muokattavissa, ja se mahdollistaa tapahtumien luonnin sekä niitten siirtämisen raahaamalla ja pudottamalla. Kalenteri on mahdollista asentaa hakemaan tapahtumat Google Calendar -kalenterista, taikka suoraan esimerkiksi MySQL-tietokannasta JSON syötteen avulla. FullCalendar:in voi ladata ilmaiseksi sen kotisivuilta, ja se sisältää mukanaan kaikki tarvittavat JavaScript-kirjastot. Kuviossa 14 on varausjärjestelmän kalenteritoiminto joka on tehty käyttäen FullCalendar:ia.

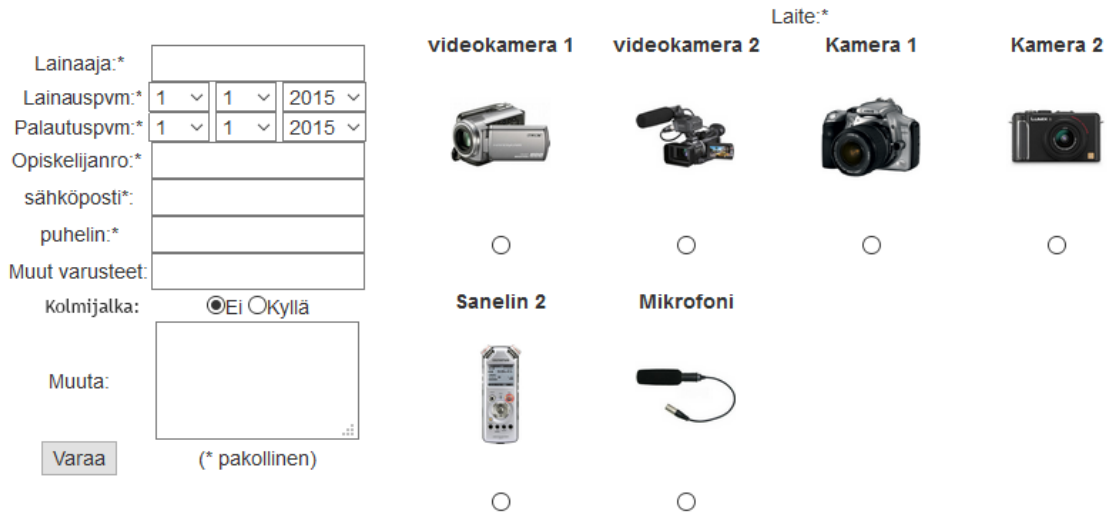


Kuvio 14: Kalenteritoiminto

Varausjärjestelmässä kalenterin avulla ei voi tehdä suoraan varauksia, sen tehtävä on vain näyttää tehdyt laitevaraukset, jotka se saa tietokannoista JSON-syötteen avulla. Järjestelmään kalenteritoiminto on asennettu näyttämään vain viikonpäivät maanantaista perjantaihin, hakemaan varaukset taulukosta laitteiden varauspäivät, sekä näyttämään varauksista lisätietoa, kun varausta painetaan. Varauksien värit on asetettu vaihtelemaan laitetyyppien mukaan.

8.5 Varaustoiminto

Varaustoiminto on järjestelmän keskeisin ominaisuus, ja kaikki muut toiminnot on luotu joko tukemaan sitä, taikka mahdollistamaan se. Varaustoiminto koostuu kolmesta eri komentosarjasta. HaeLaiteNimet-komentosarja hakee kaikki olemassa olevat laitteet tietokannan laitteet- ja laitteet_henkilöstö-taulukoista, sekä näyttää niille asetetut kuvat. DateValidation-komentosarja tarkistaa onko laite vapaana valitulle aikajaksolle, ja estää uuden varauksen luonnin, jos varaus löytyy aikajaksolle. LuoVaraus-komentosarja tarkistaa onko pakolliset kentät täytetty oikein. Kun kentät ovat täytetty oikein, sekä valitun laitteen ollessa vapaana varaa-painike luo varauksen syöttämällä varauksen varaukset-taulukkoon. Opiskelija käyttäjäryhmälle näkyvä versio varaustoiminnosta näkyy kuviossa 15.



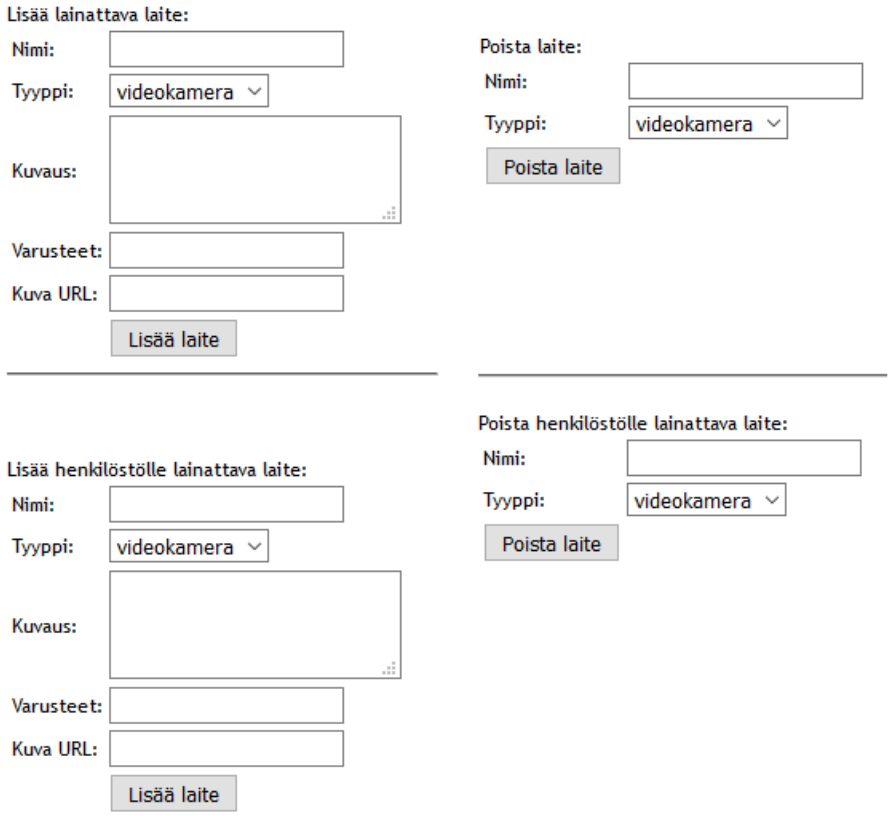
Kuvio 15: Varaustoiminto opiskelijalle

Varauksen tekemiseksi käyttäjän tarvitsee valita lainattava laite, sekä täyttää lomakkeesta tähdellä merkityt kentät. Havainnoinnissa huomattiin, että varaukseen tulee vaatia yhteystiedoiksi käyttäjän sähköpostiosoite sekä puhelinnumero. Käyttäjät eivät aina palauta laitteita ajallaan, taikka ne jopa saatetaan palauttaa väärään paikkaan, jolloin puhelinnumero ja sähköpostiosoite auttavat laitteiden sijainnin tiedustelussa.

8.6 Lisää- ja Poista-ominaisuudet

Lisää- ja poista-ominaisuudet sijaitsevat kummatkin niille tarkoitetuilla sivuillaan, ja kummastakin toiminnosta on kaksi erillistä versiota. Ensimmäiset versiot kohdistuvat tavallisiin laitteisiin, joita lainataan kaikille käyttäjille, ja näiden tietoja säilytetään laitteet-taulukossa. Toiset versiot on luotu erityislaitteita varten jotka ovat vain henkilöstölle lainattavia, ja joitten tietoja säilytetään laitteet_henkilosto-taulukossa. Toiminnot olisi voitu toteuttaa myös valintapainikkeella, josta käyttäjä valitsisi kumman käyttäjäryhmän laitteita hallitsisi. Pidin kuitenkin erillisiä versioita parempana vaihtoehtona. Erillisten versioiden tarkoitus on vähentää laitteiden hallitsemista vääristä taulukoista vahingossa.

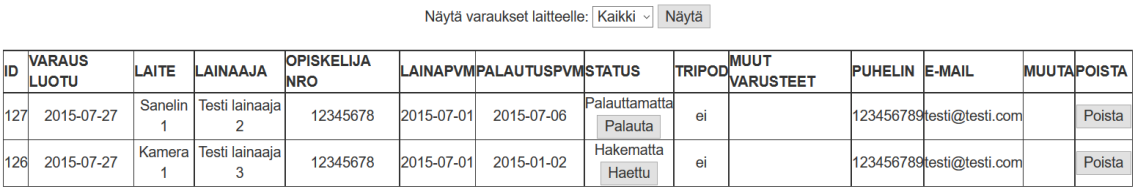
Lisätäkseen laitteen käyttäjän tarvitsee täyttää tiedot lomakkeen kenttiin jotka näkyvät kuviossa 16. Kuva URL -kenttään täytetään WWW-osoite lisättävän laitteen kuvan sijainnista, joka näytetään varaus- ja laitetiedot-sivulla. Poista laite-ominaisuudessa käyttäjän tarvitsee täyttää lomakkeeseen laitteen nimi, sekä valita laitteen laitetyyppi. Jotta laitteiden vahingossa poistaminen olisi hankalampaa, ominaisuus tehtiin sisältämään kaksi kenttää. Tietokannasta löytyvän tiedon pitää täsmätä laitteen nimen ja laitetyypin kanssa, jotta laite poistetaan järjestelmästä.



Kuvio 16: Lisää ja poista ominaisuudet

8.7 Varauksien hallinta

Varauksien hallinta on järjestelmänvalvojille tarkoitettu ominaisuus, ja se on suunniteltu näyttämään varauksien tiedot Kuvion 17 mukaisesti. Varauksien hallinnassa hallitaan myös laitteiden tilaa, sekä sieltä voi poistaa varauksia. Toiminto löytyy varausjärjestelmän pääsivulta järjestelmänvalvoja-käyttäjäryhmälle.



Kuvio 17: Varauksien hallinta

Oletuksena varauksien hallinta näyttää kaikki tietokannoista löytyvät varaukset, mutta varauksien näyttämistä on myös mahdollista rajata koskemaan vain tiettyä laitetta. Kuviossa 17 näkyvä status-sarake on suunniteltu varauksen tilan näyttämiseen, joita järjestelmästä löytyy 3, sekä niitten hallintaan. Varauksien mahdolliset tilat järjestelmässä ovat hakematta, palauttamatta, sekä palautettu.

9 Yhteenveto ja johtopäätökset

Opinnäytetyön teoriaosuudessa käydään läpi käytettävyyttä ja käyttökokemusta lyhyesti käsitteinä, sekä huomioon otettavia asioita verkkosivujen suunnittelussa jotka vaikuttavat näihin. Opinnäytetyön teoriaosuudessa käydään läpi lyhyesti toimintatarinaa ja käyttötarinaa työkaluina, jotka auttavat verkkosivujen suunnittelussa ja suunnitelman testaamisessa. Teoriaosuudessa käydään läpi myös muutamaa eri ohjelmistoa, joista on apua verkkopalvelun toteuttamisessa, sekä ohjelmointikieliä joita opinnäytetyön projektissa käytettiin.

Opinnäytetyössä toteutettiin sähköinen varausjärjestelmä Laurea-Ammattikorkeakoulussa toimivalle Studio Oivalle, joka lainaa laitteistoaan Laurean opiskelijoille sekä henkilöstölle. Sähköisen varausjärjestelmän tarkoitus on helpottaa varauksien tekemistä, niitten seuraamista, sekä mahdollistaa varauksien tekeminen etänä.

Opinnäytetyön käytännön osuudessa käydään läpi projektin kulkua, joka alkoi vaatimusmäärittelyllä. Vaatimusmäärittelyn jälkeen suoritettiin havainnointia tutkijan ollessa työharjoittelussa Studio Oivassa, jonka avulla pystyttiin kartoittamaan varausjärjestelmän tarpeita paremmin. Seuraava askel projektissa oli uuden ja vanhan toimintatavan kuvaaminen tarinoita käyttäen. Toimintatarina saatiin varauspalvelua käyttäneeltä opiskelijalta, ja sillä kuvattiin vanhaa puutteellista toimintatapaa. Käyttötarina rakennettiin toimintatarinan pohjalta, ja se antoi mahdollisuuden ilmaista ja testata suunnitelmaa varausjärjestelmästä. Seuraavaksi projektissa prototypoitiin varausjärjestelmän toiminnallisuutta, ja prototyyppejä testattiin käyttäjällä. Hyvän prototyypin tekemisen jälkeen kuvattiin sivuston rakennetta rautalankamallin avulla. Rautalankamalli toimi pohjapiirustuksena sivustolle jonka pohjalta sivusto rakennettiin.

Opinnäytetyössä käydään läpi sähköisen varausjärjestelmän toiminnallisuuksia, sekä sivuston ja tietokantojen rakennetta. Projektin viimeisiin vaiheisiin kuuluivat järjestelmän testaus ja julkaisu. Sivuston julkaisun jälkeen olin vielä Studio Oivassa työharjoittelussa noin puolivuotta, jolloin ylläpidin sivustoa, ja seurasin järjestelmän vaikutuksia Oivaan.

Tavoite työssä oli toteuttaa toimiva ja käyttäjäystävällinen varausjärjestelmä Studio Oivalle. Arvioin tavoitteen toteutuneen, vaikkakin järjestelmästä jäi hyödyntämättä osa toteutetuista käyttäjäryhmistä. Suurin ongelma työssä oli selkeästi aikataulutus, opinnäytetyön valmistuminen venyi noin vuoden käytännön osuuden valmistumisen jälkeen. Opinnäytetyötä varten olisi kannattanut alusta asti aikatauluttaa jatkuvasti aikaa, myös kirjoittamistyötä varten.

Lähteet

Kirjalähteet

Dooley, J. 2011. Software Development and Professional Practice. Apress: New York.

Gilmore, W., J. 2008. Beginning PHP and MySQL: From Novice to Professional. Kolmas painos. New York: Apress.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2007. Tutki ja kirjoita. 13. painos. Helsinki: Tammi.

Kananen, J. 2014. Toimintatutkimus kehittämistutkimuksen muotona: miten kirjoitan toimintatutkimuksen opinnäytetyönä. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Korpela, J., K & Linjama, T. 2005. Web-suunnittelu. Jyväskylä: Docendo.

Krug, S. 2006. Älä pakota minua ajattelemaan: Tervettä järkeä verkkosuunnitteluun. Ketola, V-P. Helsinki: Readme.fi.

Kriegel, A. 2011. Discovering SQL: a hands-on guide for beginners. E-kirja. Indianapolis: Wrox

Larsen, R. 2013. Beginning HTML and CSS. E-kirja. Wrox.

Nielsen, J. & Loranger, H. 2006. Prioritizing web usability. Berkeley: New Riders.

Rantala, A. 2002. PHP: Web-ohjelmoinnin peruskirja. Jyväskylä: Docendo.

Sinkkonen, I., Nuutila, E. & Törmä, S. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. Helsinki: Tietosanoma.

Sähköiset lähteet

Agile Manifesto. 2001. Manifesto for Agile Software Development. Viitattu 26.7.2016. http://agilemanifesto.org/principles.html

Apache Friends. 2016. XAMPP Installers and Downloads for Apache Friends. Viitattu 27.7.2016. https://www.apachefriends.org/index.html

Jyväskylän yliopisto. 2015. Määrällinen tutkimus. Viitattu 3.4.2016. https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/maarallinen-tutkimus

Jyväskylän yliopisto. 2015. Laadullinen tutkimus. Viitattu 3.4.2016. https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/laadullinen-tutkimus

Lotz, M. 2013. Waterfall vs. Agile: Which is the Right Development Methodology for Your Project? Viitattu 11.7.2016 http://www.seguetech.com./waterfall-vs-agile-which-is-the-right-development-methodology-for-your-project/

MariaDB. 2016. About MariaDB. Viitattu 9.7.2016. https://mariadb.com/kb/en/mariadb/about-mariadb/

Open Source Initiative. The Open Source Definition. Viitattu 29.9.2016. https://opensource.org/osd

PHP. PHP: What is PHP? – Manual. Viitattu 29.9.2016.

http://php.net/manual/en/intro-whatis.php

Saaranen-Kauppinen, A. & Puusniekka, A. 2006. 3.3.1 Validiteetti. Viitattu 1.4.2016. http://www.fsd.uta.fi/menetelmaopetus/kvali/L3_3_1.html

Saaranen-Kauppinen, A. & Puusniekka, A. 2006. 3.3.2 Reliabiliteetti. Viitattu 2.4.2016. http://www.fsd.uta.fi/menetelmaopetus/kvali/L3_3_2.html

Sarja, J. 2006. Relaatiotietokanta. Viitattu 29.9.2016. http://verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku0375c6.html?C:D=419702&selres=419702

SQLCourse. SQLCourse – Lesson 1: What is SQL?. Viitattu 29.9.2016. http://www.sqlcourse.com/intro.html

Tilastokeskus. Laadullisen ja määrällisen tutkimuksen erot. Viitattu 4.4.2016. https://www.stat.fi/virsta/tkeruu/01/07/

Vilmunen, T. 2015. Projektihallinnan vesiputousmenetelmä vs ketterät menetelmät. Viitattu 11.7.2016. http://4dsoftware.fi/projektinhallinnan-vesiputousmenetelma-vs-ketterat-menetelmat/

W3Schools. CSS Introduction. Viitattu 29.9.2016. http://www.w3schools.com/css/css_intro.asp

W3Schools. Introduction to HTML. Viitattu 29.9.2016. http://www.w3schools.com/html/html_intro.asp

Kuviot

Taulukot