



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

ILMOITUSTAULUJEN SÄHKÖISTÄMINEN

Web-pohjainen sovellus

TEKIJÄ/T: Tuomas Heikura

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Tuomas Heikura	
Työn nimi Ilmoitustaulujen sähköistäminen, web-pohjainen sovellus	
Päiväys 27.04.2020	Sivumäärä/Liitteet 32
Ohjaaja(t)	
Toimeksiantaja/Yhteistyökumppani(t) Asunto-osakeyhtiö Kuopion Hiihtäjätie 15, Eero Heikkinen	
Tiivistelmä <p>Projektin tavoitteena oli suunnitella netin välityksellä toimiva sovellus asunto-osakeyhtiön rappukäytävälle tuleviin sähköisiin ilmoitustaulu -näyttöihin osoitteessa Hiihtäjätie 15. Suunnitteluvaiheessa sovellusta päätettiin lähteä kehittämään web -pohjaisena. Suurimpina kriteereinä oli saada sovelluksesta helppokäyttöinen ja aikaa kestävä.</p> <p>Sovellusta lähdettiin kehittämään WAMP-ympäristössä, joka koostuu useammasta ohjelmasta ja muodostaa web -palvelimen. Varsinaista sovelluksen koodia kirjoitettiin Notepad++ -sovelluksella. Teknologiana käytettiin muun muassa HTML, PHP, JavaScript, CSS, sekä erilaisia ohjelmointikirjastoja. Sovelluksen toimivuutta testattiin ilmaisella InfinityFree -nimisellä web-hotellilla.</p> <p>Projektilla saatiin luotua toimiva kokonaisuus, joka on asunto-osakeyhtiön hallittavissa ja interaktiivinen myös rappukäytävien asukkaille. Sovelluksen ilmoitustaulu -osio tulee näkyviin asunto-osakeyhtiön investoimiin näyttötauluihin, joita olisi tarkoitus saada jokaiseen rappukäytävään asukkaiden nähtäville.</p>	
Avainsanat Tietotekniikka, web-sovellus, ilmoitustaulu	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Tuomas Heikura			
Title of Thesis Web-Based Application for the Message Board Electrification			
Date	27 April 2020	Pages/Appendices	32
Supervisor(s)			
Client Organisation /Partners Housing Company Kuopion Hiihtäjätie 15, Eero Heikkinen			
<p>Abstract</p> <p>The aim of the thesis was to design a web-based application for the electronic message board screens on the stairwells of Housing Company at Hiihtäjätie 15. At the planning phase, it was decided to start developing the application as a web-based application. The biggest criteria were to make the application easy to use and time-lasting.</p> <p>The application was developed in a WAMP environment, which consists of several applications and forms a web server. The actual application code was written using Notepad++. The used technologies were HTML, PHP, JavaScript, CSS and various programming libraries. The application was tested on a free web hotel called Infinity-Free.</p> <p>As a result of this thesis, a functional entity was created that is under the control of Housing Company and interactive for the residents of the stairwells. The message board section of the application will be shown on the display panels invested by Housing Company.</p>			
Keywords Information technology, web-program, message board			

SISÄLTÖ

1	JOHDANTO	6
1.1	Käsitteet.....	7
2	KÄYTETYN TEKNOLOGIAN ESITTELY	8
2.1	WampServer.....	9
2.2	Ohjelmointikielät ja tekniikat.....	9
2.2.1	HTML	9
2.2.2	CSS.....	11
2.2.3	JavaScript.....	12
2.2.4	jQuery.....	13
2.2.5	Bootstrap.....	13
2.2.6	PHP.....	14
2.2.7	MySQL.....	16
3	WEB-SOVELLUKSEN SUUNNITTELU	17
3.1	Varhaissuunnitelma	17
3.2	Aloituspalaveri ja suunnittelun syventäminen	19
3.2.1	Taloyhtiön käyttöliittymä	19
3.2.2	Asukkaiden käyttöliittymä	19
3.2.3	Ilmoitustaulun näkymä	20
4	WEB-SOVELLUKSEN KEHITTÄMINEN.....	21
4.1	Alkuvaihe	21
4.2	Tietokanta.....	21
4.3	Rappukäytävien ilmoitustaulu	23
4.4	Asukkaiden varaus- ja ilmoittautumisnäkyvä	25
4.4.1	Responsiivisuus.....	25
4.4.2	Päivämäärän valitsin.....	26
4.5	Taloyhtiön hallintänäkyvä	28
4.5.1	Asukastietojen hallinta.....	29
4.5.2	Ilmoitusten hallinta	31
4.5.3	Varausten hallinta	32
4.5.4	Ponnahdusikkunat.....	34
5	POHDINTAA INFONÄYTÖISTÄ	35

5.1	Vaatimukset ja hintataso	35
6	POHDINTA JA JATKOKEHITYS	36
6.1	Pohdinta.....	36
6.2	Jatkokehitysideoita.....	37
7	KUVAT.....	38
8	LÄHTEET	39

1 JOHDANTO

Opinnäytetyöni aiheena oli lähteä suunnittelemaan ja toteuttamaan asunto-osakeyhtiön kohteeseen Hiihtäjätie 15 sähköistä ilmoitustaulu -järjestelmää. Heidän suunnitelmissa oli tuoda heidän rappukäytäviinsä sähköiset näyttöpäätteet, joihin hakivat tietotekniikan opiskelijaa suunnittelemaan ja toteuttamaan ilmoitustaulusovellusta.

Opinnäytetyön aihe saatiin kouluni opinto-ohjaajan kautta. Työntilaajan yhteyshenkilö Arttu Rautiainen lähestyi opiskelijoita Savonia -ammattikorkeakoulun nettisivuilta löytyvällä ”Tarjoa työtä opiskelijalle” -lomakkeella. Heidän hakemuksessaan tuli ilmi, että heidän tavoitteensa oli sähköistää rappukäytäviensä ilmoitustaulut.

Kyseessä on siis toiminnallinen opinnäytetyö, tarkemmin mainittuna kehittämistyö. Aiheen kiinnostavuutta lisäsi se, että siinä annettiin vapaudet työntekijän käsiin liittyen sovelluksen toteutustapaan ja -tekniikoihin.

Sovellusta lähdettiin toteuttamaan web-pohjaisena, jolloin se olisi helposti päivitettävissä ja käytettävissä myös etäisesti taloyhtiön ja asukkaiden laitteilta. Sovellusta aloitettiin suunnittelemaan WAMP-ympäristössä, joka muodostaa web-palvelimen kotikoneelle paikallisena. Sovelluksen luomiseen käytettiin seuraavanlaisia web-ohjelmointiin liittyviä teknologioita HTML5, PHP, JavaScript, CSS, jQuery, MySQL ja BootStrap.

Opinnäytetyön tuotoksen pääasiallinen tarkoitus on nopeuttaa, sekä helpottaa taloyhtiön ja asukkaiden vuorovaikutusta toisiinsa. Tavoitteena on saada helppokäyttöinen, toimiva ja aikaa kestävä lopputuote, joka otettaisiin käyttöön asunto-osakeyhtiön kohteessa. Oppimistavoitteena on kehittyä opiskelijana web-ohjelmoinnissa aikaisempaa laajemmin ja syvemmin. Erityisesti tietokanta suunnittelussa ja siihen liittyvässä ohjelmoinnissa.

1.1 Käsitteet

HTML (Hypertext Markup Language) = Kuvauskieli, jota käytetään web-sivustojen luomiseen erilaisia html -elementtejä hyödyntäen.

PHP (Hypertext Preprocessor) = Palvelinperäinen ohjelmointikieli, jolla saadaan web-sivustoille dynaamisuutta ja vuorovaikutteisuutta.

CSS (Cascading Style Sheets) = Tyylistyvä, joiden avulla HTML -koodattu dokumentti saadaan näyttämään tyyliään sellaiselta kuin itse halutaan.

JavaScript = Web-ohjelmointiin tarkoitettu kieli, joka tuo sivustoille dynaamista toiminnallisuutta.

jQuery = Yksi JavaScript kielen helposti ymmärrettävissä oleva kirjasto, jonka avulla saadaan paljon toiminnallisuutta vähemmällä koodimäärällä.

Bootstrap = Avoimen koodin kehitysympäristö, joka sisältää CSS-elementtejä ja JavaScript-osa.

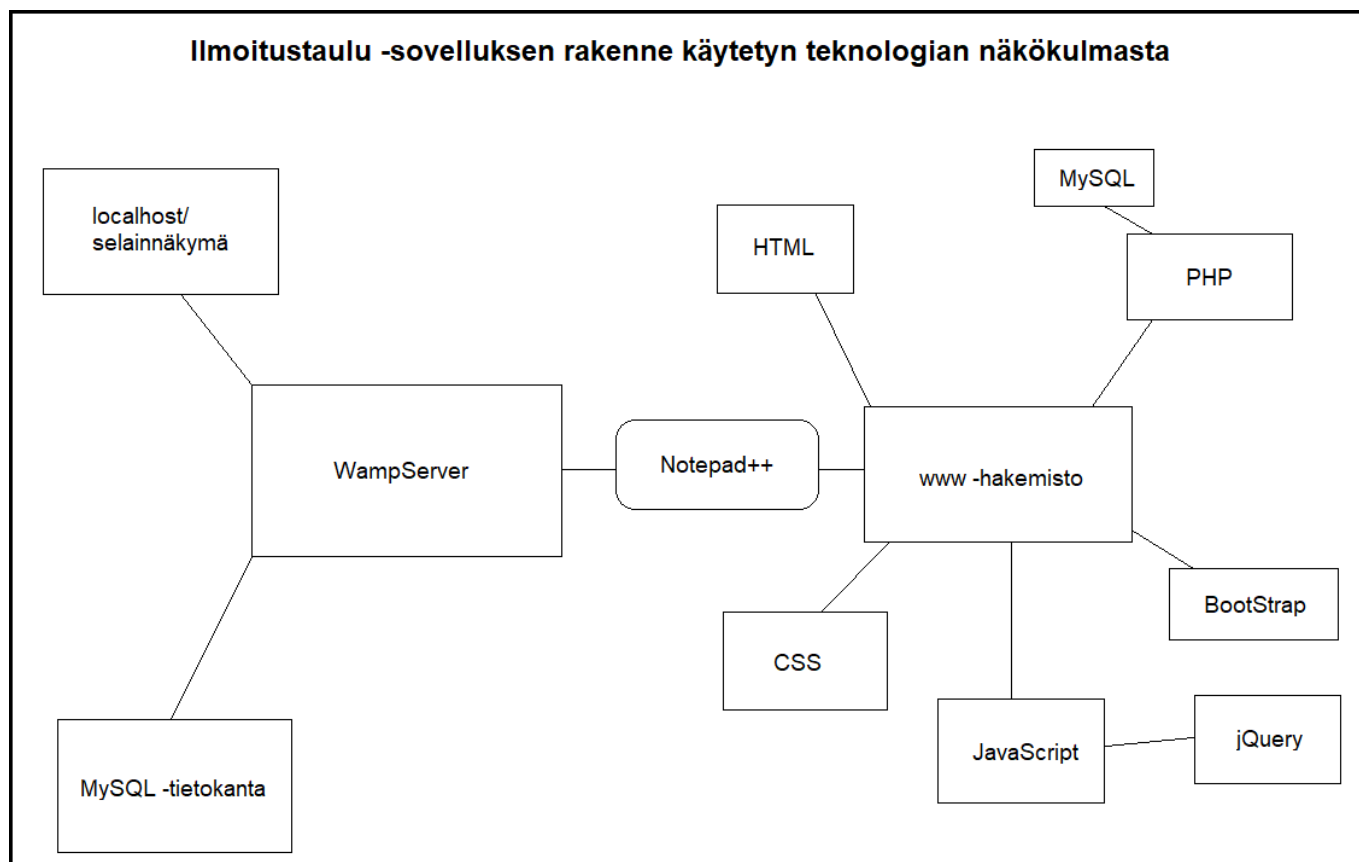
MySQL = Relaatiotietokantaohjelmisto, jota käytetään esimerkiksi erilaisten tietojen varastointiin.

WampServer = Windows-pohjainen verkkokehitysympäristö, joka sisältää useamman ohjelmiston ja muodostaa web-palvelimen.

Skriptikieli = Toiselta nimeltään komentosarjakieli, jonka ohjelma tulkkaa lennosta ilman tarvetta kääntää sitä etukäteen. Esimerkiksi PHP ja JavaScript.

2 KÄYTETYN TEKNOLOGIAN ESITTELY

Tässä luvussa käydään läpi tarkemmin ilmoitustaulu -sovelluksen suunnitteluun ja luomiseen käytettyjä ohjelmointi välineitä ja teknologiaa. Ensiksi esitellään WampServer, jonka muodostama palvelinympäristö toimii suunnittelu- ja luomisympäristönä koko projektin ajan. Esitellään myös sovelluksessa käytetty tietokanta, sekä ohjelmointikielien ja tekniikoiden. Ohjelmointikielistä käydään lyhyesti läpi niiden esittely, historia ja kuvien avulla muutama esimerkki.



Kuva 1. Sovelluksessa käytetty teknologia

2.1 WampServer

Lyhenne WAMP tulee sanoista Apache2, PHP, MySQL ja koska tätä ympäristöä ylläpidetään Windows -käyttöjärjestelmällä saa lyhenne tästä ensimmäisen kirjaimensa. WampServer on verkkokehitykseen tarkoitettu Windowsilla toimiva alusta, joka auttaa sinua luomaan tehokkaita web-sovelluksia. Kuten nimestä voidaan päätellä, WampServerin yhteydessä käyttöösi tulee HTTP -palvelinohjelma Apache2, PHP -ohjelmointikieli, sekä tietokantajärjestelmä MySQL tai MariaDB. WampServer sisältää paljon toiminnallisuutta, voidaan muunmuassa hallita Apache ja MySQL -palveluita, sallia pääsy myös ulkopuolisilta, käsitellä palvelimen asetuksia ja tarkastella loki- ja asetustiedostoja. WampServer on ladattavissa ilmaiseksi ja sitä löytyy kahtena eri versiona, 32- ja 64-bittisenä. (Bourdon, ei pvm)

2.2 Ohjelmointikielien ja tekniikat

2.2.1 HTML

HTML-lyhenne tulee sanoista HyperText Markup Language, joka on web-ohjelmoinnissa käytetty kuvauskieli, jolla suurimmat osat netin sivustoista on kuvattu. Sitä käytetään nettisivustojen luomiseen ja mahdollistamaan niiden toiminnallisuuden. HTML sana lyhenne HT eli Hypertext tarkoittaa, että sivusto sisältää linkkejä, jotka sallivat käyttäjän siirtyä paikasta toiseen samalla sivustolla tai siirtyä toisiin sivustoihin. Lyhenne ML eli Markup Language on tietokoneiden tapa kommunikoida toisilleen kontrolloidakseen sitä, kuinka teksti on käsitelty ja esitetty. Tähän HTML tarvitsee tageja ja määritteitä, jotka ovat sen perusta. (Moraes, 2020)

Kuvassa 2 nähdään HTML -koodilla tehty hyvin yksinkertainen pohja, jossa myös esimerkkejä tageista ja selitys mitä määritteet tarkoittavat. Pelkästään HTML -koodia käyttämällä sivustojen ulkoasu jää hyvin yksinkertaiseksi, tämän asian korjaamiseksi käytetään lisäksi myös CSS -tyylisivuja, jotta sivustosta saadaan näyttävä myös visuaalisesti. CSS:sää käydään läpi tarkemmin myöhemmässä vaiheessa tätä kappaletta.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Sivun otsikko</title>
5 </head>
6 <body>
7
8 <h1>Tämä on otsikko (header) -tagi.</h1>
9 <p>Tämä on kappale (paragraph) -tagi.</p>
10
11 <p align="right">Tämä on kappale -tagi, jossa on määritteenä (attribute) kohdistus oikealle.</p>
12
13 </body>
14 </html>
15

```

Tämä on otsikko (header) -tagi.

Tämä on kappale (paragraph) -tagi.

Tämä on kappale -tagi, jossa on määritteenä (attribute) kohdistus oikealle.

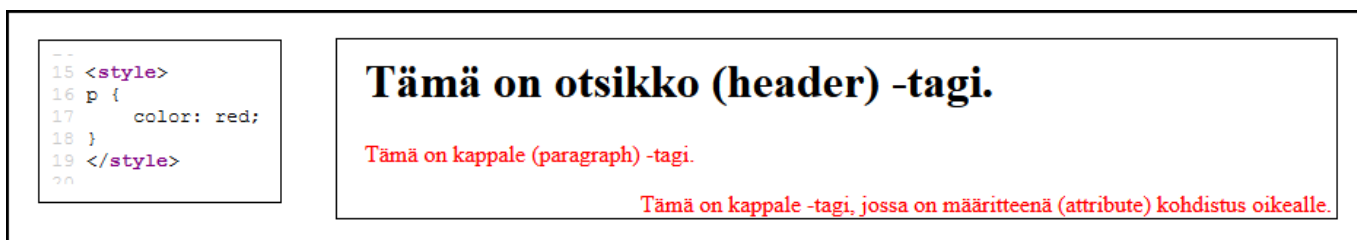
Kuva 2 Esimerkki HTML -pohjasta, alapuolella selainnäkyä.

HTML:n historia yletty 90-luvun alkuun saakka. Sen keksijänä pidetään CERN:n työntekijää Sir Tim Berners-Lee:tä, joka julkaisi ensimmäisen Web -dokumentin nimeltään HTML Tags vuonna 1991. HTML yritettiin ehdottaa viralliseksi standardiksi ensimmäisen kerran vuonna 1993, mutta vasta 1995 IETF järjestämä HTML-työryhmä sai julkaistua HTML 2.0, josta tuli ensimmäinen virallinen HTML-standardi. Seuraava HTML 3.2 -versio julkaistiin vuonna 1997 toisen standardiorganisaation W3C:n kehittämänä. Tämä versio sisälsi viimeisimpiä edistyksiä, kuten Java -sovelmien ja kuvien ympärillä leijuvia tekstejä. (Delgado, 2019)

Suurena harppauksena edellisiin HTML -versioihin verrattuna 4.0 -versio julkaistiin vuonna 1998, joka toi mukanaan muunmuassa CSS -tyylitaulukot, mahdollisuuden sisällyttää sivustolle pieniä ohjelmia tai skriptejä eli komentosarjoja, parannuksia suunniteltujen sivujen helppokäyttöisyydessä, monimutkaiset taulukot ja monia parannuksia lomakkeissa (forms). Virallisista HTML -määritelmistä viimeisin julkaitiin vuonna 1999, joka oli nimeltään HTML 4.01. Tämän jälkeen HTML:n standardointitoiminta pysähtyi, joka herätti Apple-, Mozilla- ja Opera -yhtiöiden kiinnostuksen perustaa uusi WHATWG -niminen yhdistys. Nykyään tämä ryhmän toiminta keskittyy HTML5 -standardiin, josta julkaistiin ensimmäinen virallinen luonnos vuonna 2008. (Delgado, 2019)

2.2.2 CSS

CSS eli Cascading Style Sheets tarkoittaa tyylisivuja, joiden avulla voidaan kuvata HTML- tai XML -koodilla kirjoitetun dokumentin esittelyä eli tyylitellä. CSS -tyyliohjeilla voit käyttää tyylejä valikoivasti eri HTML -dokumenttien elementteihin. CSS:llä voit esimerkiksi muuttaa tekstin fonttia, väriä, kokoa ja rivivälejä, jakaa sen useampaan sarakkeeseen tai lisätä animaatioita ja muita koristeellisia ominaisuuksia. CSS on yksi avoimen verkon keskeisimmistä tyylittelyyn käytetyistä tekniikoista ja on standardisoitu kaikilla selaimilla W3C -organisaation määritelmän mukaan. (MDN-contributors, 2020)



Kuva 3 Esimerkki CSS -tyylittelystä, kuvassa oikealla selainnäkömä.

Kuvassa 3 on esimerkkinä, kuinka CSS:llä voidaan vaihtaa esimerkiksi kappaleen fontin väriä lyhesti lisäämällä tämä esimerkiksi HTML-dokumentin tyyli -elementtien sisälle.

CSS johtaa juurensa vuoteen 1994, jolloin CERN:n työntekijä Håkon Wium Lie huomasi työskennellessään yrityksessä nimeltä MIT Media Laboratory, että verkko tarvitsee tyylilohe -kieltä. Lie kokeili CSS -sivustoa ensimmäisen kerran Arena -nimisellä selaimella, jonka jälkeen jatkoi CSS1, CSS2 ja RFC 2318 -versioiden kehitystä yhdessä Tim Berners-Leen ja Robert Cailliaun kanssa. CSS:stä tuli vuosien 1994-2004 aikana vakiintunut verkkostandardi, joka vaikuttaa edelleenkin suuresti maailmanlaajuisen verkon ulkoasuun ja helppokäyttöisyyteen. Uusin ja nykyisin CSS3 -versio julkaistiin vuonna 1999. Nykyisin W3C:n työryhmä pyrkii jatkuvasti parantamaan CSS:ää, he lisäävät jatkuvasti uusia, pienempiä komponentteja nykyiseen version erilaisten laajennusten kautta. CSS4 versiota ei siis ole julkaistu version CSS3:n jälkeen ja monet väittävätkin, ettei edes CSS3 ole olemassa, vaan niillä pikemminkin määritellään päivitykset, jotka ovat tapahtuneet CSS2 -version jälkeen. (SimpleCSS, ei pvm)

2.2.3 JavaScript

JavaScript on web -ohjelmoinnissa käytettävä nimensä mukainen skriptikieli eli komentosarjakieli. Se on tulkittu kieli eli se ei tarvitse ohjelmointikielen kääntäjää toisin kuin esimerkiksi ohjelmointikielien C tai C++. JavaScriptillä kirjoitettu kieli toimii siis suoraan selaimessa. Web -sovelluksia ja verkkosivuja rakentaessa HTML:n ja CSS:n lisäksi sivustoille on hyvä saada myös jonkinlaista toiminnallisuutta ja siksi JavaScriptillä on tärkeä rooli näiden kolmen ohjelmointikielen keskuudessa web-ohjelmoinnissa. Javascript tuo sivustoille ja sovelluksille dynaamisen toiminnan, sen komentosarjat ohjaavat erilaisia funktioita, kun painikkeita napsautetaan, kuinka salasaneläkkeitä todennetaan tai miten mediaa hallitaan. HTML, CSS ja JavaScript toimivat harmoniassa keskenään luodakseen täysimittaisia sovelluksia. (Grant, 2019)

JavaScript:n toiminnassa on kaksi tärkeää asiaa, selaimen toiminta ja DOM (Document Object Model) eli dokumenttioliomalli. Ensiksi selain lataa verkkosivun, jäsentelee HTML -koodin ja luo tästä dokumenttioliomallin. Seuraavaksi DOM esittelee sivuston live-näkymän JavaScript komentosarjalle. Tämän jälkeen selain tarttuu kaikkeen HTML:n dokumenttiin linkitettyyn sisältöön, kuten kuviin ja CSS -tiedostoihin. Ennen kuin selaimen JavaScript-moottori lataa komentosarjatiedostot ja sisäisen koodin, dokumenttioliomalli on koontanut HTML ja CSS luodakseen verkkosivuston. JavaScript-moottori ei kuitenkaan suorita komentosarjoja heti vaan se odottaa HTML:n ja CSS:n latautuvan ensimmäisenä, jonka jälkeen suorittaa koodinsa kirjoitusjärjestyksessä. Tämän seurauksena JavaScript päivittää dokumenttioliomallia ja selain tulkitsee sitä. Jos Javascript ei odottaisi HTML:n ja CSS:n latautumista, se ei voisi muuttaa DOM -elementtejä. (Grant, 2019)

Brendan Eich kehitti JavaScript:n vuonna 1995 työskennellessään Netscape Communications -nimisessä yrityksessä, joka hallitsi markkina-asemaa sen ajan parhaimmalla verkkoselaimella, Netscape Navigator:lla. Tarkoituksena oli saada helppo, HTML:n sekaan sijoitettava ja hetkessä tulkittavissa oleva kieli. Hänelle annettiin huhujen mukaan kaksi viikkoa aikaa, mutta jo puolentoista viikon kuluessa Eich oli luonut heikosti toimivan prototyypin, joka kuitenkin päätyi pidemmiten tuotantoon uuden Netscape Navigator 2.0 -version kanssa. Vaikka tämä komentosarjakoodi tunnetaan nimeltä JavaScript niin se on kuitenkin virallisesti ECMAScript. JavaScript:stä on kehitelty vuosien saatossa monia kirjastoja, joita ovat esimerkiksi AJAX ja jQuery, joita käydään läpi tarkemmin myöhemässä vaiheessa raporttia. (Forsström, 2015)

2.2.4 jQuery

jQuery on yksi JavaScript:n kirjastoista eli se on joukko valmiiksi kirjoitettua JavaScript -koodia. jQuery:n avulla verkkokehittäjä voi jättää pitkien ja tavanomaisten JavaScript -koodipätkien kirjoittamisen pois ja keskittyä sellaisiin monimutkaisiin ominaisuuksiin, joita tulee näkymään vain heidän sivustollaan. jQuery tuo mukanaan myös monenlaisia muita hyötyjä verkkokehittäjälle, se on avointa lähdekoodia eli kenen tahansa muokattavissa ja sillä on suuri käyttäjäkunta, jolloin sitä tuetaan jatkuvasti. jQuery on myös laajasti dokumentoitu ja se toimii myös kaikkien muidenkin JavaScript -kirjastojen kanssa. Toiminallisuus on myös helposti laajennettavissa, koska jQuery:llä on tuhansia liitännäisiä. (Morris, Skillscrush, 2020)

jQuery:n syntyhistoria ylettyy kymmenen vuoden päähän JavaScript:n luomisesta, vuoteen 2005, jolloin John Resig kehitti sen alkuperäisen version ja julkaisi seuraavana vuonna BarCampNYC-nimisessä tapahtumassa. John Resig halusi JavaScript -koodin kirjoittamisen olevan hauskaa, jonka jQuery toteutti tekemällä toistuvista tehtävistä lyhyitä, älykkäitä ja ymmärrettävissä olevia poistamalla ylimääräiset merkinnät. jQuerystä oli tavoitteena tarjota ergonominen ohjelmointirajapinta verkkosivujen käsittelyyn, sekä poistaa selainten väliset erot, joka oli tuona aikana vielä vaikeaa. Myöhemmin jQuery:n kehittyminen toi mukanaan uusia tekniikoita, kuten animaatiot ja Ajax, joka on joukko erilaisia web-sovelluskehitys tekniikoita. jQuerysta tuli niin suosittu, että se hallitsee edelleen valtaosaa internetistä. W3Tech:n arvion mukaan noin 74 prosenttia kaikista verkkosivuista löytyy jQuery -tekniikkaa. (Guo, 2019)

2.2.5 Bootstrap

Tänä päivänä verkkosivustojen kehityksessä responsiivisuus on tärkeä asia eli sivustot ovat skaalautuvissa kaikille päätelaitteille. Bootstrap on suosituin avoimen lähdekoodin kehys tällaiseen tarkoitukseen, joka on myös ilmainen. Se pitää sisällään HTML-, CSS- ja JavaScript -komponentteja erilaisien ominaisuuksien luomiseen. Bootstrapin avulla näiden kielten koodimäärä saadaan pudotettua huomattavasti vähemmäksi kuin ilman sitä. Sivustojen responsiivisuus jouduttiin aiemmin toteuttamaan CSS Media Queries -palvelulla, joka oli Bootstrap:ia monimutkaisempi ja hitaampi. (Yogihosting, 2020)

Bootstrap oli alkuperäiseltä nimeltään Twitter Blueprint ja sen loivat Mr. Mark Otto ja Jacob Thorton. Bootstrap kehitettiin twitterissä ja julkaistiin avoimena lähdekoodina GitHub:ssa 2011. "GitHub on verkkosivusto- ja pilvipohjainen palvelu, joka auttaa kehittäjiä varastoimaan ja hallitsemaan koodejaan sekä seuraamaan ja kontrolloimaan muutoksia niihin" (Kinsta, 2020). Bootstrapilla oli tarkoitus lisätä web-sivustojen sisäisten työkalujen yhdenmukaisuutta ja jättää pois ulkoiset kirjastot, jotka olivat epäjohdonmukaiset ja vaativat raskasta ylläpitoa. Bootstrap toi mukanaan paljon erilaisia hyötyjä, kuten kaikkien selainten tuki, keveys, web-kehittäjän muokattavissa oleva ja yksinkertaisuus sekä tehokkuus. (w3school, ei pvm)

2.2.6 PHP

HTML:n, CSS:n ja JavaScript:n avulla päästään jo hyvin alkuun, kun suunnitellaan ja toteutetaan web -pohjaista sovellusta. Projektin ilmoitustaulu -sovellus onkin pääosin tehty näillä kolmella tekniikalla, mutta koska sovellukseen liittyy palvelin, tarvitaan myös palvelinpuolen ohjelmointikieltä. Tähän tarkoitukseen otettiin käyttöön PHP -kieli, koska se on tarpeeksi riittävä tähän projektiin.

PHP eli Hypertext Preprocessor on skriptikieli, jota käytetään web-ohjelmoinnissa palvelinpuolella. Vastakohtana palvelinpuolen kielille on asiakaspuolen tekniikat, joita ovat esimerkiksi edellämainitut HTML, CSS ja JavaScript. Asiakaspuolella tarkoitetaan tässä tilanteessa paikallisesti käyttäjän tietokoneen selaimen kautta tapahtuvia toimintoja ja se on puoli, joka on näkyvässä käyttäjälle. Käyttäjältä piilossa oleva palvelinpuoli taas toimii siten, että selain lähettää pyynnön palvelimelle, joka vastaa tähän HTML -muodossa ja selain näyttää tämän käsittelynsä jälkeen käyttäjälle. Yhteenvedottuna asiakaspuolen komentosarjakielellä, kuten JavaScript:llä tuotettu sisältö on heti saatavilla, kun taas palvelinpuolen sisältö, joka on kirjoitettu PHP:llä vaatii ensiksi pyynnön palvelimelta tai tietokannasta. Sisällön jako myös palvelinpuolelle tuo sivustoihin tehokkuutta, latausnopeutta ja tallennuskapasiteettia. PHP:n käyttäminen web-kehityksessä tuo sivustoille täysin uuden tason dynaamisuutta JavaScript:n lisäksi. (Morris, Skillcrush, 2020)

PHP on yleisin web-ohjelmoinnissa käytetty kieli, jota käyttää noin 82% kaikista verkkosivustoista. PHP:n keksi Rasmus Lerdorf vuonna 1994 ja julkaisi siitä ensimmäisen versionsa vuonna 1995 nimellä PHP/FI. Tuohon aikaan PHP oli vain yksinkertainen c-kielellä kirjoitettu komentosarjatyökalu, joka kuitenkin kehittyi nopeasti PHP3 -version julkaisun jälkeen vuonna 1998. Tämä versio sisälsi useita edistyneitä ominaisuuksia, kuten Zend -skriptimoottorin. PHP:stä on julkaistu vuosien varrella ensimmäisen version jälkeen 15 erilaista päivitystä. Uusin näistä on versio 7.4, joka julkaistiin marraskuussa vuonna 2019 (PHP.net, 2019). Nyky päivänä PHP:tä käyttävät useimmat isommat yritykset, kuten Facebook, Wikipedia, Yahoo ja WordPress verkkossa olevien porfolioidensa käsittelyyn. (Nawaz, 2019)

Kuvassa 4 nähdään yksinkertainen esimerkki PHP-koodista, jossa selainnäkömään tuodaan tekstiä. Kuva liittyy aiemmissa esimerkeissä käytettyihin kuviin, lisäyksenä PHP-koodi.

```
<?php  
echo "Terve maailma!";  
?>
```

Tämä on otsikko (header) -tagi.

Tämä on kappale (paragraph) -tagi.

Tämä on kappale -tagi, jossa on määritteenä (attribute) kohdistus oikealle.

Terve maailma!

Kuva 4 Esimerkki PHP-koodista.

2.2.7 MySQL

MySQL on avoimen lähdekoodin relaatiotietokannan hallintajärjestelmä (RDBMS), jota käytetään tietokantojen luomiseen ja hallitsemiseen perustuen relaatiomalliin. Tietokannalla tarkoitetaan erilaisista tiedoista koostuvaa joukkoa ja koska MySQL -järjestelmän tietokannat perustuvat relaatiomalliin, tiedot on järjestetty taulukoiksi. Jokaisella taululla on jonkinlainen yhteys tietokannassa. MySQL -hallintajärjestelmällä on asiakas-palvelin -malli. Asiakkaalla tarkoitetaan tässä yhteydessä tietokoneita, joissa tällaista RDBMS -järjestelmä suoritetaan. Palvelimella taas tarkoitetaan RDBMS-palvelinta, johon asiakastietokone ottaa yhteyden käyttäessään tietokannan tietoja. (Dwika, 2020)

MySQL on yksi suosituimmista RDBMS -ohjelmistoista. Sitä käyttävät muunmuassa Facebook, Twitter, Youtube ja Google. Se oli alunperin rajoitetussa käytössä, mutta nykyisin se on yhteensopiva muunmuassa Linux:n, macOS:n, Microsoft Windows:n ja Ubuntu:n kanssa. (Dwika, 2020)

MySQL sanan perässä oleva SQL lyhenne tulee sanoista Structured Query Language eli jäsenneily kyseylikieli, jolla palvelin ja asiakas kommunikoivat. Muita SQL-syntaksia käyttäviä hallintajärjestelmiä ovat esimerkiksi PostgreSQL ja MicrosoftSQL-palvelin. MySQL itsessään on kuitenkin kirjoitettu C- ja C++ -ohjelmointikielillä. SQL -kielellä kirjoitetuilla lausekkeilla saadaan palvelin suorittamaan erilaisia toimintoja. Niillä voidaan esimerkiksi kysyä tietoa tietokannasta, käsitellä tietoja, kuten poistaa, lisätä, muuttaa tai lajitella, määritellä tietotyyppelijä ja hallita käyttöoikeuksia tietokantaan tallennetuille tiedoille. (Dwika, 2020)

Vuonna 1995 perustettu ruotsalainen yritys nimeltä MySQL AB loi MySQL:n. Kehitystiimiin kuuluivat Michael Widenius, David Axmark ja Allan Larsson. MySQL -projekti sai alkunsa jo vuonna 1979, jolloin Michael Widenius kehitti tietokantojen hallintaan tarkoitettun UNIREG -tietokantatyökalun. MySQL syntyi, kun Michael kysyi mSQL:n kirjoittajan David Hughesin kiinnostusta yhdistää mSQL UNIREG:iin. MySQL:n etuliite tulee Michael Wideniuksen tyttären nimestä "My". Vuonna 2000 MySQL:stä tulee avoimen lähdekoodin ohjelmisto, joka julkaistaan GPL:n ehtojen mukaisesti. Vuonna 2001 MySQL saavuttaa 2 miljoonaa aktiivista asennusta ja jatkaa kasvuaan seuraavien kahden vuoden aikana miljoonalla vuosittain. MySQL käynnistää MySQL-verkon vuonna 2005, joka tarjoaa loppukäyttäjille muunmuassa päivityksiä, hälytyksiä, ilmoituksia ja tukea. Samana vuonna julkaistaan uusi versio MySQL 5, joka sisältää paljon uusia ominaisuuksia. Vuonna 2006 Oracle yrittää ostaa MySQL:n. Aktiivisten asennusten määrä nousee samaisena vuonna 8 miljoonaan. Vuonna 2008 Sun Microsystems ostaa MySQL AB -yrityksen miljardilla dollarilla. Kahden vuoden jälkeen Oracle ostaa Sun Microsystems:n 7,4 miljardilla dollarilla ja siirtyy näin MySQL:n omistajaksi. (Jaiswal, 2018)

3 WEB-SOVELLUKSEN SUUNNITTELU

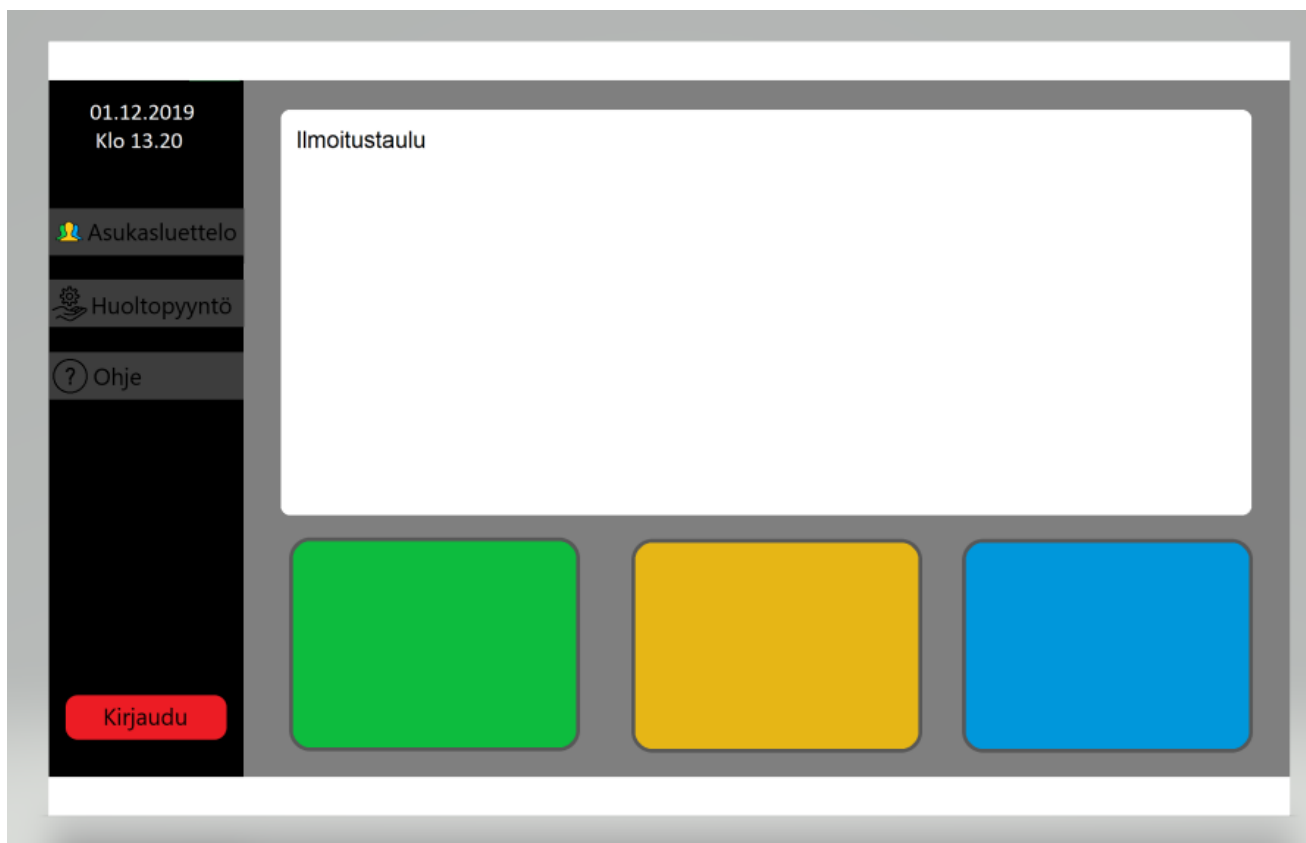
Tässä luvussa käydään tarkemmin läpi sitä, miten tämä projekti sai alkunsa, minkälaisia vaatimuksia sovellukselle asetettiin ja miten sitä alettiin suunnittelemaan. Suunnittelu koostui useammasta osasta. Aluksi varhaiset suunnittelut luotiin piirtämällä näyttökuvia tulevalle ilmoitustaululle. Aloituspalaverin jälkeen saatiin vaatimukset, joiden pohjalta aloitettiin varsinaisen sovelluksen suunnittelu. Projektin edetessä osa suunnitelmista muuttui ja kehittyi parempaan suuntaan.

3.1 Varhaissuunnitelma

Aihe saatiin koulun opettajan lähettämän viestin kautta, jossa etsittiin koodaajaa suunnittelemaan ja toteuttamaan rappukäytävien ilmoitustaulujen sähköistäminen. Aihe vaikutti kiinnostavalta, koska itseltäni löytyy vastaavaa kokemusta aiemmista koulussa tehdyistä projekteista. Sähköisistä ilmoitustauluistakin löytyi käyttäjäkokemusta ja olen huomannut, että ne ovat paljon kätevämpiä kuin klassiset ilmoitustaulut.

Aiheen valitsemisen jälkeen tiedossa ei ollut muuta kuin, että sovellus olisi web-pohjainen ja siinä olisi tarkoitus saada näkymään vaihtuvaa ajankohtaisasiaa ja -sisältöä. Sovellus päädyttiin suunnittelemaan selainpohjaiseksi ja käyttämään tekniikkana yleisimpiä web-ohjelmointi -tekniikoita, kuten HTML, CSS, JavaScript, koska näistä löytyy itseltäni vahvaa kokemusta aiemmista vastaavista projekteista ja koska aikataulullisesti oli järkevää jättää kokonaan uusien ohjelmointitekniikoiden opettaminen pois. Tiedossa ei myös ollut vielä sitäkään, tuleeko näyttötauluissa olemaan kosketustoimintoa vai ei. Varhaista suunnitelmaa lähdettiin tekemään piirtämällä erilaisia näyttökuvia, joihin haettiin inspiraatiota jo markkinoilla olevista ilmoitustaulu -sovelluksista.

Kuvassa 5 on ensimmäinen näyttökuva, joka oli tarkoitettu kosketusnäytölliseen ilmoitustauluun. Tämä näyttökuva suunniteltiin antamaan vain jonkinlaista suuntaa projektille, vaikka tarkempia vaatimuksia ei vielä ollut tiedossa.



Kuva 5 Suunnitteluvaiheen ensimmäinen näyttökuva

3.2 Aloituspalaveri ja suunnittelun syventäminen

Aloituspalaveri pidettiin tammikuun alussa, jolloin kirjoiteltiin sopimukset opinnäytetyön toteuttamisesta asunto-osakeyhtiölle. Palaverissa käytiin läpi tarkempia vaatimuksia ja selvisi, että tämä ilmoitustaulusovellus tulisi toimimaan näytöissä, joissa ei ole kosketusmahdollisuutta. Koska sovellukselle vaadittiin jonkinlaista interaktiivisuutta, joka ei onnistuisikaan suoraan ilmoitustaulun näyttöä kosketellen, lähdettiin tätä suunnittelemaan aivan uudesta näkökulmasta.

Ilmoitustaulun sovelluksen tavoitteena oli tuoda asukkaille informaatiota, mahdollistaa kommunikointi asukkaiden ja taloyhtiön välillä, sekä muistuttaa myös tavallista, alkuperäistä ilmoitustaulua. Sovellusta täytyi laajentaa alkuperäisestä suunnitelmasta siten, että siinä olisi oma käyttöliittymä niin taloyhtiölle kuin asukkaille, sekä näkymä rappukäytävässä olevalle näyttöpäätteelle. Sovellus vaatisi myös jonkinlaista tiedon varastointia eli tietokannan suunnittelu oli myös edessä.

3.2.1 Taloyhtiön käyttöliittymä

Taloyhtiön käyttöliittymä tulisi sisältämään asukkaiden tietojen hallitsemisen eli uusien asukkaiden lisäämisen, muokkaamisen, poistamisen ja esikatselun. Käyttöliittymässä olisi myös mahdollisuus hallita erilaisia talon tilojen varauksia, kuten sauna- ja pyykkivuorot. Admin-käyttäjien eli taloyhtiön kuuluvan henkilön käytössä olisi siis varauksien tarkastelu ja tarvittaessa poistaminen. Yksi ilmoitustaulun vaatimuksista oli näyttää kiinteää ja vaihtuvaa sisältöä, joita Admin-käyttäjien olisi mahdollisuus luoda ja poistaa tässä käyttöliittymässä. Kiinteässä sisällössä tulisi näkymään esimerkiksi taloyhtiön järjestyssääntöjä ja vaihtuvassa taas erilaisia taloyhtiön järjestämiä tapahtumia tai muita ajankohtaisia ilmoitusasioita.

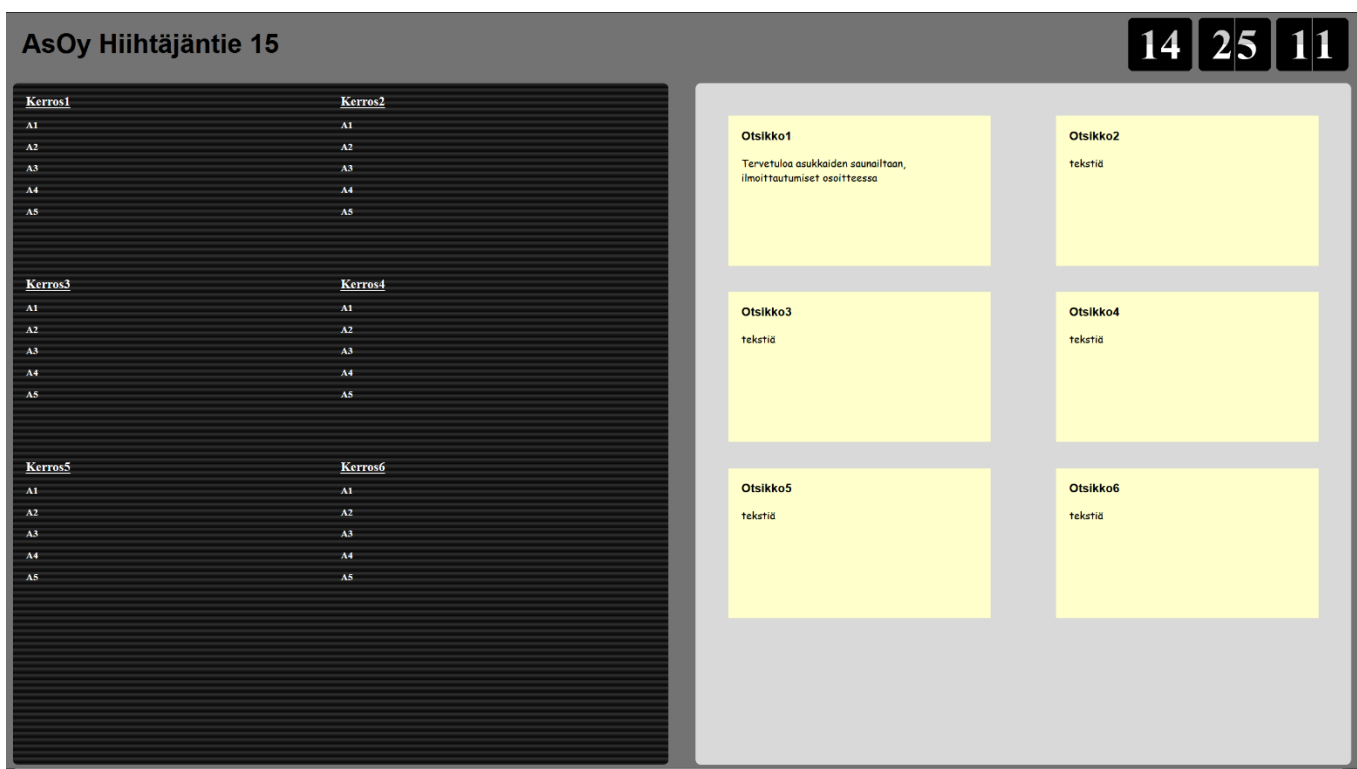
3.2.2 Asukkaiden käyttöliittymä

Asukkaiden käyttöliittymä tulisi olemaan ensisijaisesti suunniteltu toimimaan mobiili-laitteissa, mutta tarkoituksena tehdä koko web-sovelluksesta responsiivinen, jotta se toimisi asiallisesti myös muissakin laitteissa. Käyttöliittymästä olisi tarkoitus saada mahdollisimman yksinkertainen käyttää ja siinä tulisi olemaan ominaisuuksiltaan ainakin yhteisten tilojen varausmahdollisuus ja omien varauksien hallitseminen, sekä ilmoitusten tekeminen taloyhtiölle. Asukkaan todentaminen tulisi tapahtumaan jonkinlaisella kirjautumismahdollisuutena ja jokaisella asukkaalla olisi oma henkilökohtainen kirjautumistunnus.

3.2.3 Ilmoitustaulun näkymä

Ilmoitustaulun näkymästä yritettiin saada mahdollisimman alkuperäisen näköinen, jossa kuitenkin olisi lisäksi kaikenlaista ylimääräistä digitaalisuutta, kuten päivämäärän ja kellon ajan näkyminen, sekä mahdollisesti paikallisia säätietoja. Alkuperäistä ilmoitustaulua lainaten, siinä tulisi näkymään ensisijaisesti asukkaiden sukunimet ja huoneistot kerroksineen, jotka olisi jaettu yhteen osioon. Toisessa osiossa tulisi näkymään kiinteäsisältö ja kolmannessa vaihtuvaa sisältöä.

Kuvassa 6 esitellään varhainen suunnitelma ilmoitustaulun näkymästä, jossa näyttöä on jaettu asukkaiden tietoihin ja ilmoitussisältöön.



Kuva 6 Suunnitteluvaiheen näyttökuvaa

4 WEB-SOVELLUKSEN KEHITTÄMINEN

Tässä luvussa käydään läpi ilmoitustaulu -sovelluksen kehittämistä suunnitteluvaiheen jälkeen. Luku on jaettu osioihin, joissa käydään syvemmin sovelluksen rakennetta, toiminnallisuutta ja ulkoasua. Osioissa tutustutaan sovelluksen eri käyttöliittymiin, tietokanta valintaan ja sen rakenteeseen, sekä näytetään muutamia koodipätkiä kuvineen. Tässä luvussa näytettävissä kuvissa asukkaiden nimet ovat keksittyjä ja toimivat vain suunnittelun apuna.

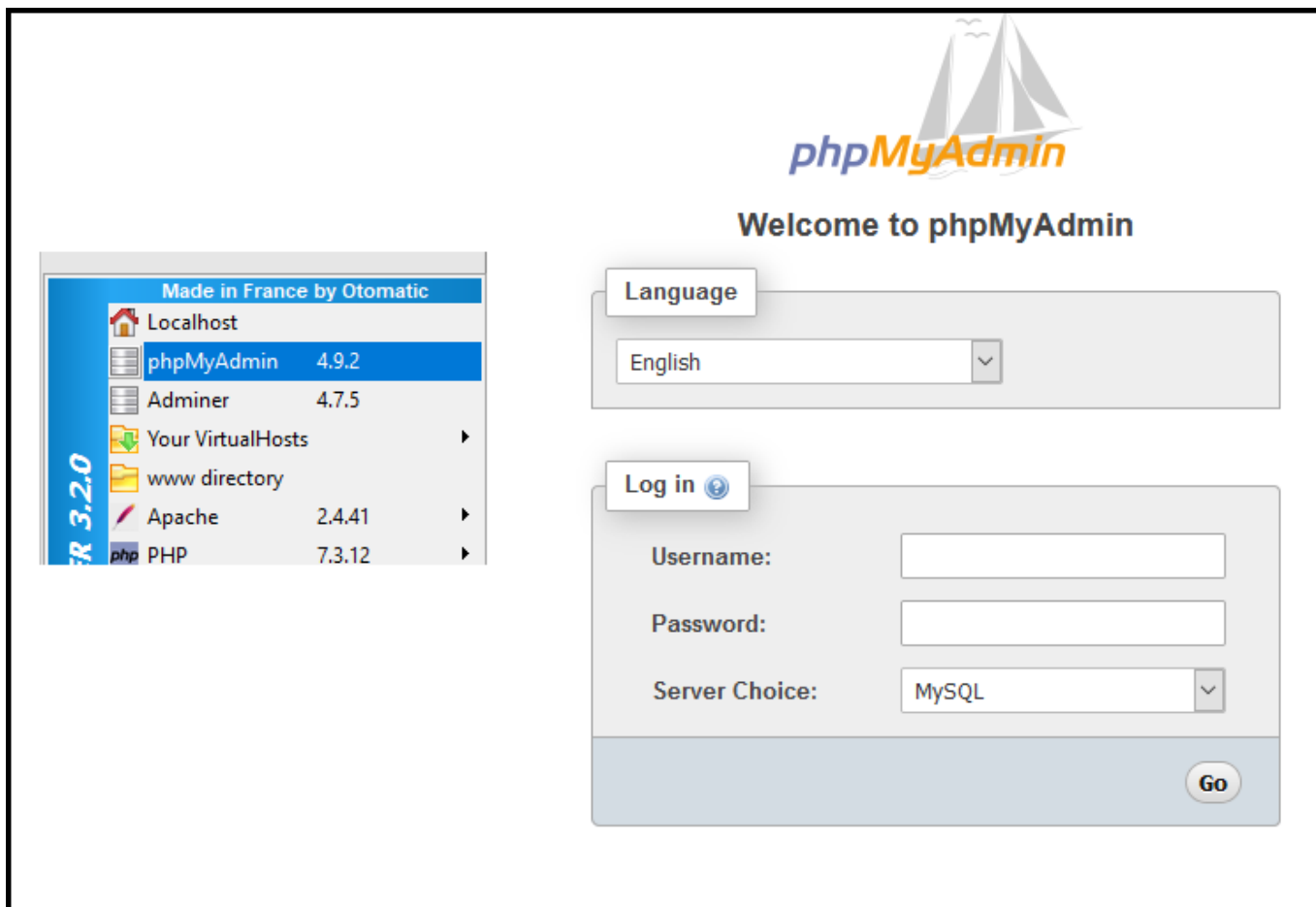
4.1 Alkuvaihe

Alkupalaverin ja suunnittelujen jälkeen aloitettiin sovelluksen kehittäminen. Ohjelmointikielien ja tekniikoiden oltiin pääpiirteittäin valittu valmiiksi. HTML:llä tulitaisiin kuvaamaan web-sovelluksen näkymää selaimen, CSS:llä muovaamaan ulkoasut sopiviksi huomioonottaen eri käyttäjät ja heidän näyttönsä koot, JavaScript:llä ja siihen liittyvillä kirjastoilla lisäämään tarvittavia toimintoja asukkaiden ja taloyhtiön käyttöönsä, PHP:llä kommunikoidaan palvelimen ja asiakaspuolen välillä ja varastoimaan tiedot MySQL -tietokantaan.

Suunnittelu- ja kehitysympäristöksi valittiin WampServer, jonka kautta on helppo esikatsella miltä kirjoitettu koodi näyttää selaimella. Koodin kirjoittamiseen valittiin Notepad++ -tekstieditori. Ensimmäisenä lähdettiin kehittämään ilmoitustaulun näkymää, josta kerrotaan tarkemmin myöhemmässä kappaleessa. Ilmoitustaulu -näkyvän ensimmäisen version hahmottamisen jälkeen aloitettiin kehittämään hallintapuolta eli taloyhtiön käyttöön tulevaa järjestelmää. Samalla aloitettiin myös asukkaille tulevan varausjärjestelmän luomista.

4.2 Tietokanta

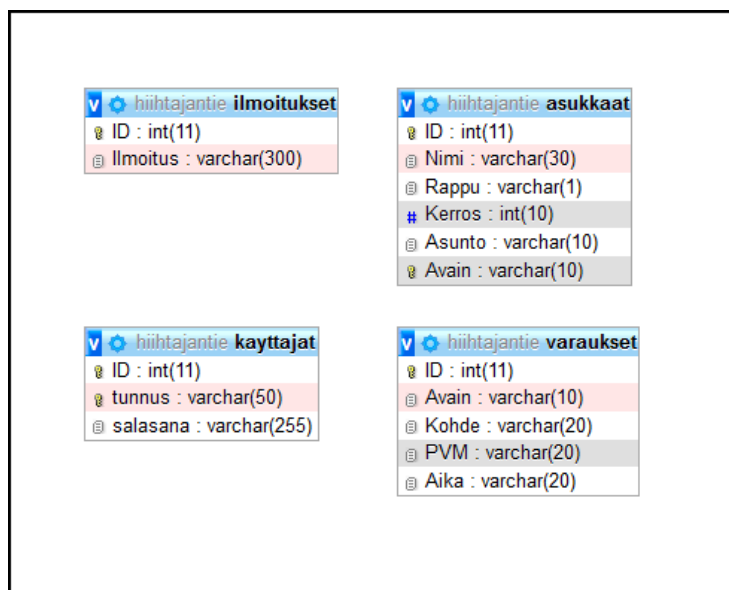
Web-sovelluksen kehitys alkoi siitä, että lähdettiin asentamaan WAMP-ohjelmistoa tietokoneelle. Koska yksi WAMP -ympäristön asentuvista ohjelmistoista oli MySQL, päädyttiin tämä valitsemaan sovelluksen tietokannaksi. Aikaisempaa kokemusta MySQL:n konfiguroinnista ja lausekkeista on itselleni tullut koulussa tehdyistä projekteista ja MySQL -kurssista. Tietokannan alustaminen ja suunnittelu onnistui kätevästi selaimen kautta käytettävän phpMyAdmin -hallintatyökalun kautta, josta näyttökuva alapuolella näkyvässä kuvassa 7.



Kuva 7 phpMyAdmin -hallintatyökalu

Tietokantaan oli tarkoitus saada ensimmäiseksi tiedot asukkaista, jotka tulisivat näkyviin ilmoitustaulu - näkymään rappukäytävälle perinteisen ilmoitustaulun tyyliin. Koska varausjärjestelmän tarvitsisi jonkinlaisen asukkaiden todentamisen, päädyttiin lisäämään asukkaiden tietokantatauluun myös tiedon asukkaiden henkilökohtaisista salasana-avaimista. Varauksille täytyi tehdä oma taulu, johon tuli tietoja muunmuassa varauksen tekijästä, ajankohdasta ja kohteesta. Erilaiset ilmoitustaululla näkyvät ilmoitukset jätettiin tietokannasta toistaiseksi pois käytöstä, koska ilmoitusten lisääminen ajateltiin toteuttaa kuvina suoraan palvelimelle. Tietokantaan tuli myös taulu sovelluksen hallintapuolen käyttäjille.

Kuvassa 8 esitellään tietokannan tietosisällön rakennetta, josta löytyy omat taulunsa asukkaille, varauksille ja käyttäjille. Ilmoitukset -taulu jätettiin vielä mahdollisen jatkokehityksen varalta.



Kuva 8 Tietokannan rakenne

4.3 Rappukäytävien ilmoitustaulu

Rappukäytävien näytöille oli tarkoitus tuoda näkyviin tiedot asukkaista, ilmoituksista ja muuta lisätietoa, kuten kellonaika ja päivämäärä. Ilmoitustauluille kehiteltiin myös QR-koodi, jonka kautta asukkaat pääsevät tekemään varauksia ja ilmoittautumaan erilaisiin taloyhtiön järjestämiin tapahtumiin omilla mobiililaitteillaan. Alkupalaverissa käydyn suunnitelman mukaan ilmoitustaululle oli myös saatava vaihtuvaa sisältöä eli ilmoitukset. Tämän päätettiin tehdä Slideshow/Carousel -tekniikalla, jossa ilmoitukset vaihtuvat tietyn ajan välein, joka on taloyhtiön käyttäjien säädettävissä ilmoituksia lisättäessä.

AsOy Hiihtäjätie 15

2020
Huhtikuu
16
13
05

Kerros1

A1 Pekkarinen
A2 Toivonen

Kerros2

A3 Matikainen
A4 Tuomainen


Kerros3

A5 Makkonen
A6 Heiskanen

1/3

Testi-ilmoitus

Yhteisten tilojen varaus:

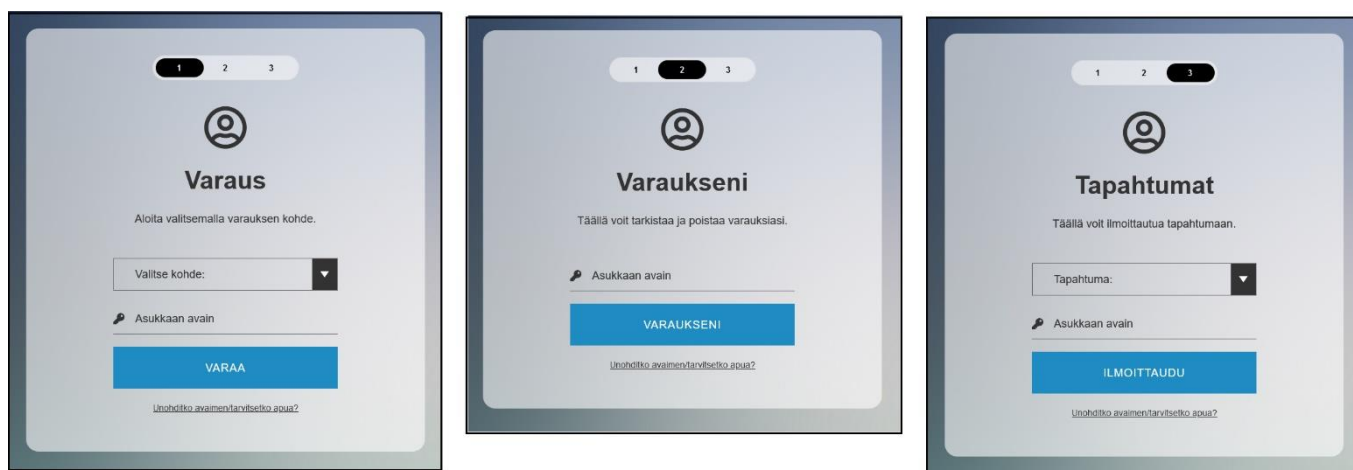


Kuva 9 Ilmoitustaulu -näkyvä

Kuvassa 9 näkyvä ilmoitustaulu on alustava näkymä, joka saattaa vielä muuttua, kunhan sovellus päätetään ottaa käyttöön ja rappukäytävien näyttökoot selviävät. Ilmoitustaulu näkymä on PHP -tiedosto, joka on pääosin kirjoitettu HTML:llä, CSS:llä ja JavaScriptillä jQuery -kirjastoa hyväksikäyttäen. Näkymän PHP-osio sisältää MySQL -kyselyn, jolla haetaan tarvittavat tiedot tietokannasta näkymään ilmoitustaululla. Ilmoitustaulusta yritin saada mahdollisimman neutraalia, silmille ystävällistä värityyliä.

4.4 Asukkaiden varaus- ja ilmoittautumisenäkymä

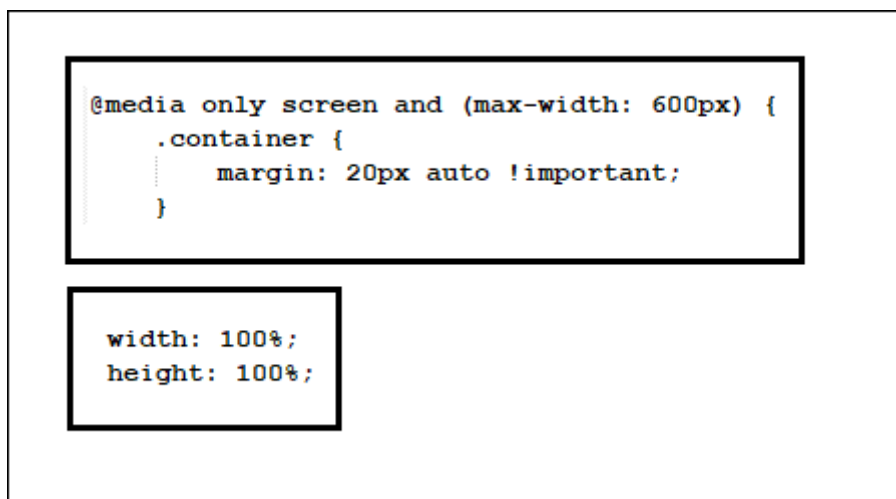
Asukkaiden käytössä oleva varausnäkömää koostuu kolmesta välilehdestä, joihin kuuluu kohteen varaaminen, omien varausten hallinta ja tapahtumiin ilmoittautuminen. Näkömää on toteutettu pääosin HTML:llä, CSS:llä ja JavaScriptillä. Mukana on myös käytetty JQuery:ä, jolla tuotiin ulkoasuun hieman lisätehostusta, kuten symboleja. Kommunikointi palvelimen tietokantaan on toteutettu PHP -koodilla ja MySQL -lausekkeilla. Kuvassa 10 on näyttökuvat varaussivuston välilehdistä.



Kuva 10 Asukkaan varausnäkömään välilehdet

4.4.1 Responsiivisuus

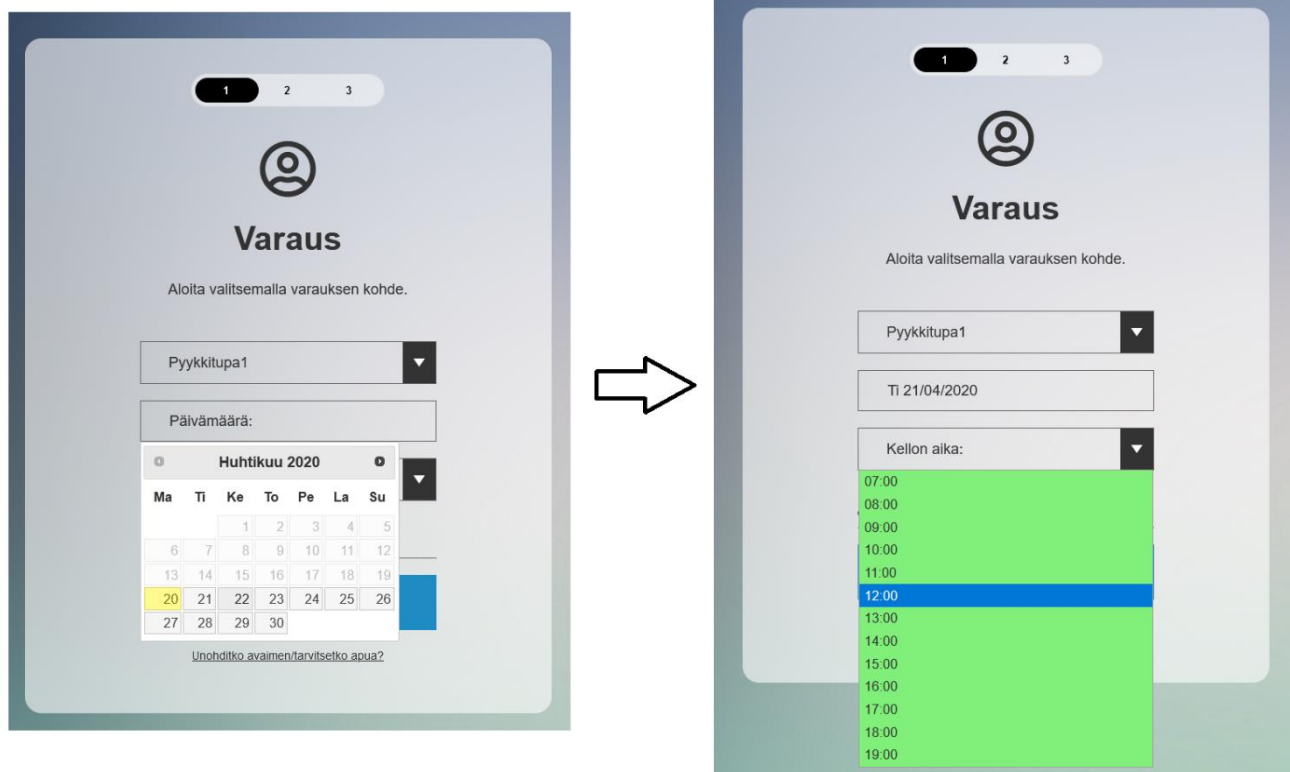
Varaussivustosta on tehty responsiivinen, joten se soveltuu hyvin asukkaiden käyttöön myös mobiililaitteilla. Responsiivisuuteen käytettiin esimerkiksi kuvassa 11 näkyviä CSS-koodin pätkiä. Ylempällä koodilla voidaan antaa erilaisia rajoitteita ja muutoksia HTML -elementeille, kun sovellusta käytävän laitteen näyttökoko muuttuu. Alemmalla koodilla vaikutetaan elementtien pituuteen ja korkeuteen. Tässä tilanteessa ei anneta mitään vakioarvoa, vaan koko ilmaistaan prosentteina, jolloin elementtien leveys ja pituus muuttuu suhteessa käytettävän laitteen näyttökokoon.



Kuva 11 Responsiivisuuden vinkkejä

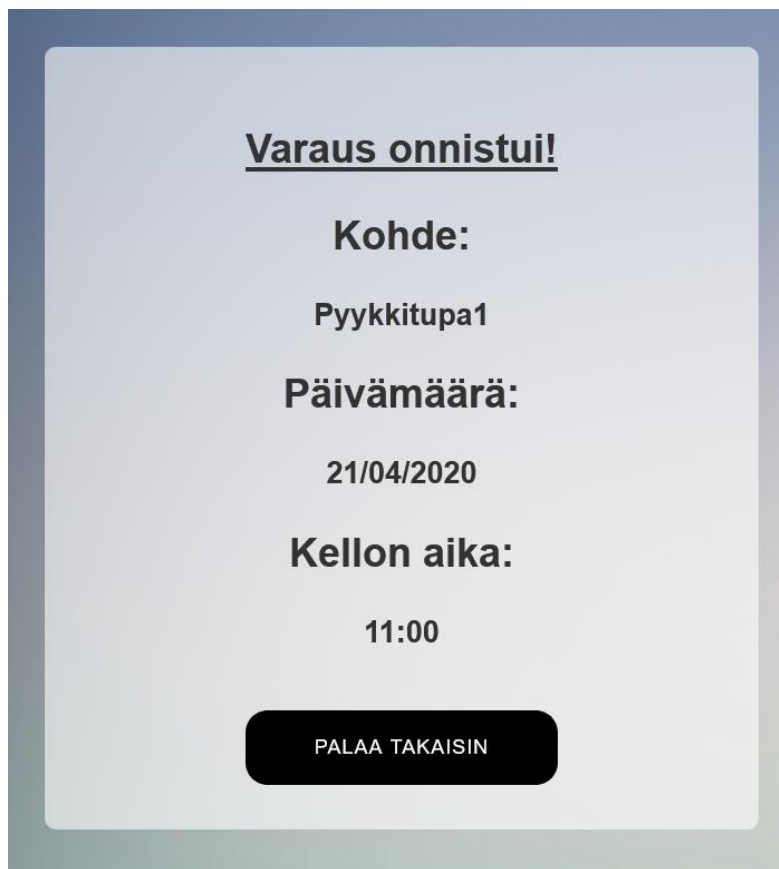
4.4.2 Päivämäärän valitsin

Varaustoiminnoissa hyödynnettiin jQueryä, joka on yksi JavaScript:n kirjastoista. Varauksen ajankohdan valitsemiseen käytettiin jQuery:n datepicker:iä. Datepicker:llä saadaan näkyviin kalenterinäkömän, josta on helppo valita varauksen päivä. Se on myös helposti muokattavissa omaan käyttöön, muunmuassa voidaan rajoittaa päivien valitsemista eli varausten tekemistä tietyille päiville. Varauskohteen valitsemisen jälkeen varaussivulle ilmestyy kuvassa 12 vasemmalla esitelty kalenterinäkömä. Ajan valitsemisen kohdalla haetaan tietokannasta tieto, onko kyseiselle päivälle vapaita aikoja. Vapaat ajat näytetään vihreällä ja mahdolliset varatut ajat punaisella. Varauksen vieminen loppuun vaatii vielä asukkaan henkilökohtaisen avaimen syöttämisen sille varattuun kohtaan.



Kuva 12 Varauksen ajankohdan valitseminen

Mikäli varaus onnistuu, se ilmoitetaan asukkaalle kuvassa 13 näkyvällä ilmoituksella, josta asukas voi vielä tarkistaa varauksen tiedot.



Kuva 13 Varauksen onnistuminen

Asukkaalla on myös mahdollisuus tarkastella ja poistaa varauksia jälkikäteen syöttämällä kirjautumisavaimensa kuvassa 10 näkyvälle Varaukseni -välilehdelle, josta avautuu kuvan 14 mukainen varausten hallinta taulukko. Asukas voi poistaa varauksensa oikealla puolella näkyvästä jätteen symbolista painamalla. Kaikkien asukkaiden varaukset ovat myös taloyhtiön hallittavissa heidän käyttäytymässään, josta kerrotaan lisää seuraavassa osiossa.

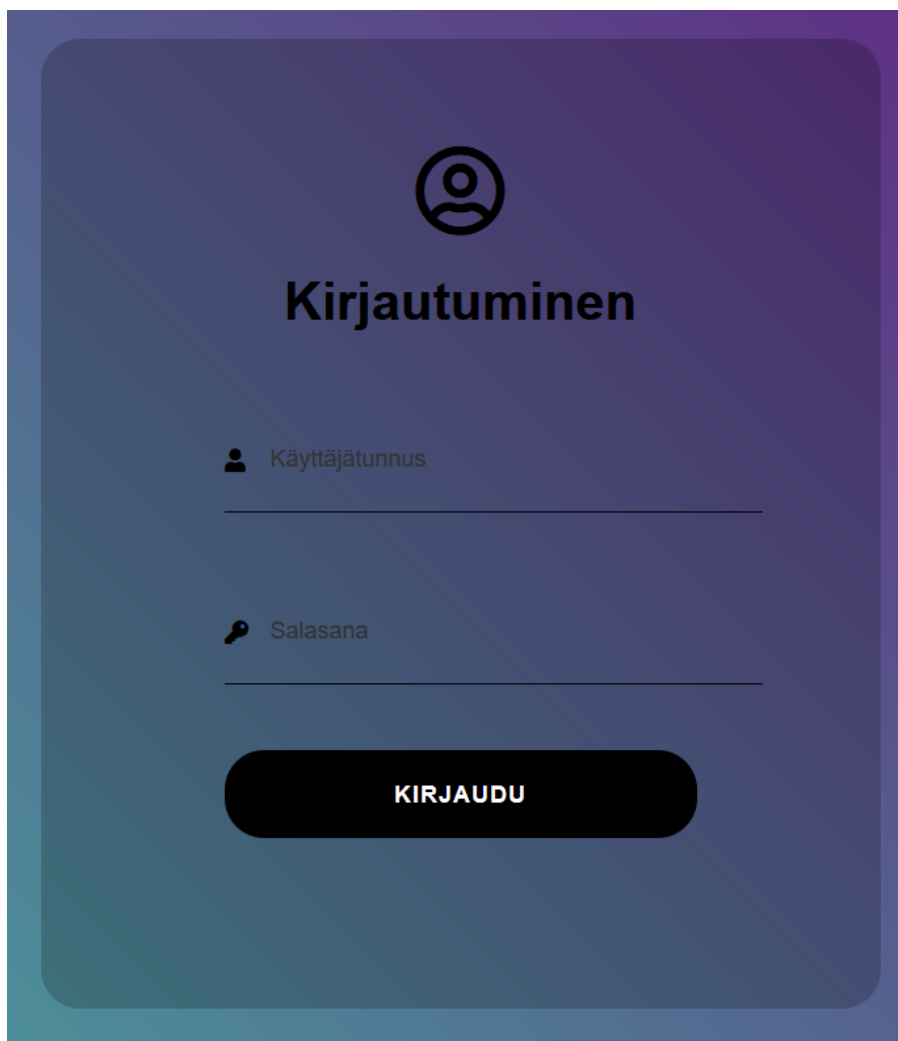
Kohde	Pvm	Aika	
Pyykkitupa2	23/02/2020	15:00	🗑️
Pyykkitupa2	24/04/2020	11:00	🗑️
Pyykkitupa1	21/02/2020	12:00	🗑️

Kuva 14 Asukkaan varausten hallinta

4.5 Taloyhtiön hallintanäkymä

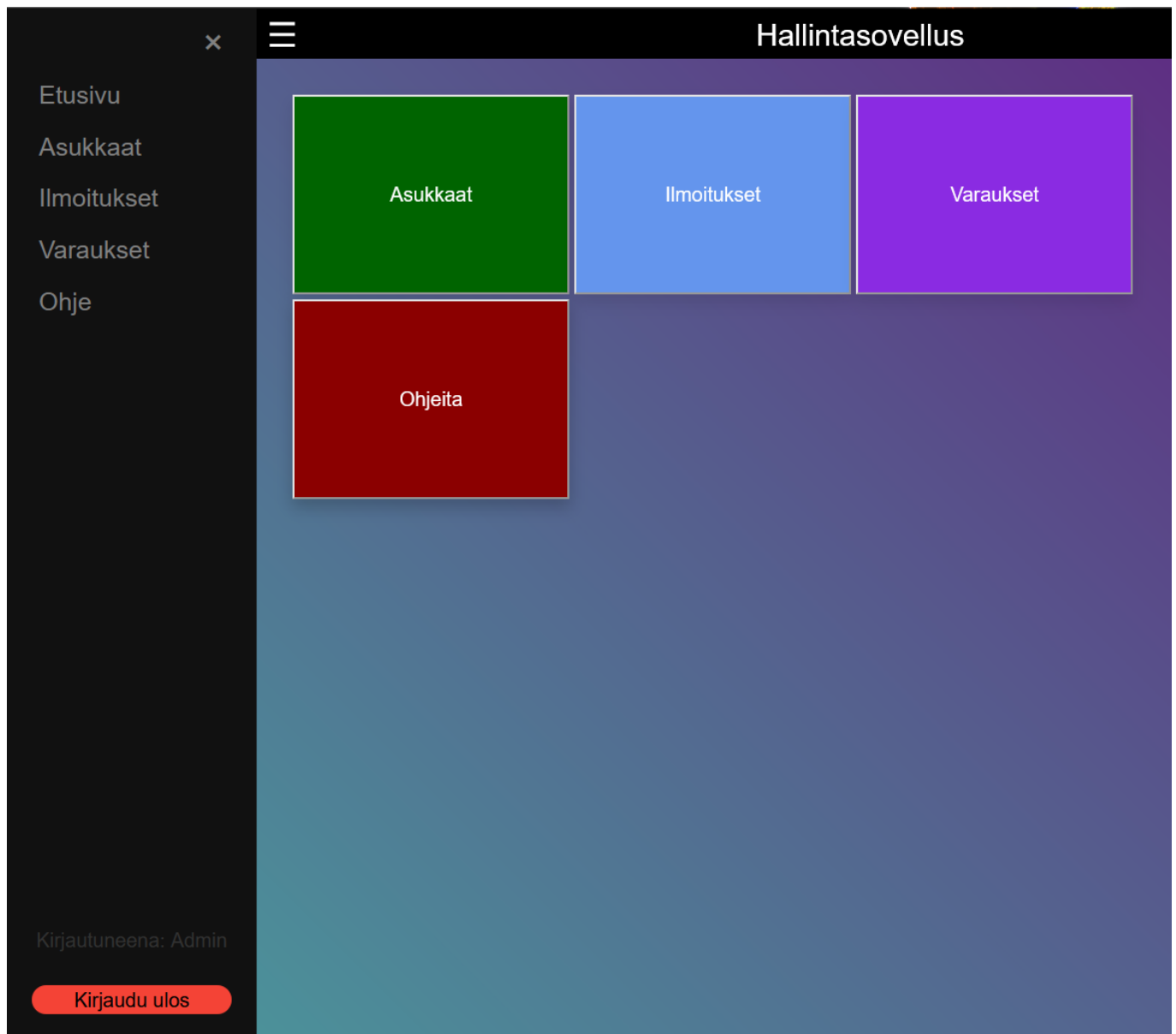
Hallintasovellukseen pääsy vaatii kirjautumisen käyttäjätunnuksella ja salasanalla. Tämä tapahtuu kirjautumissivulta, jossa kirjautumistiedot suodatetaan htmlspecialchars -funktiolla, jolloin Cross-site Scripting (XSS) hyökkäykset voidaan estää. XSS -hyökkäyksillä tarkoitetaan ulkopuolisen käyttäjän toimea, jossa ohjelmakoodia upotetaan HTML -dokumentin sekaan, jolloin selain suorittaa sen.

Kuvassa 15 on näyttökuva hallintasovelluksen kirjautumissivulta.



Kuva 15 Hallintasovelluksen kirjautumisenäkymä

Onnistuneen kirjautumisen jälkeen käyttäjälle avautuu kuvan 16 mukainen näkymä hallintasovellukseen, josta päästään hallitsemaan rappukäytävien asukkaiden tietoja, ilmoituksia ja varauksia. Etusivun vasemmasta yläkulmasta voidaan avata kuvassa näkyvä navigointi valikko, josta voidaan myös kirjautua ulos.



Kuva 16 Hallintasovelluksen etusivunäkymä

4.5.1 Asukastietojen hallinta

Asukkaiden hallinta on toteutettu mahdollisimman yksinkertaiseksi käyttää. Näkymän ylä osasta löytyy satunnaisgeneraattori, jolla voidaan luoda uudelle asukkaalle henkilökohtainen avain, jota hän käyttää varatessaan yhteisiä tiloja tai ilmoittautuessaan tapahtumiin. Asukkaiden lisääminen ja poistaminen on jätetty pois, koska asuntojen määrä pysyy aina vakiona, joten uuden asukkaan muuttaessa joihinkin asunnoista, on järkevää vaihtaa vain tämän asunnon käyttäjän nimi ja avain. Tästä syystä asunnon tietojen muokkaus on rajattu kuvassa 17 näkyvällä tavalla.

Luo avain:

Nimi	Rappu	Kerros	Asunto	Avain	
<input type="text" value="Pekkarinen Pauliina"/>	A	1	A1	<input type="text" value="XDH624FJD6"/>	<input type="button" value="Tallenna"/>
Toivonen Anni	A	1	A2	P1VDE9PXUA	<input type="button" value="Edit"/>
Matikainen Antti	A	2	A3	7H49XK5T9M	<input type="button" value="Edit"/>
Tuomainen Mauri	A	2	A4	KYIPPBK61E	<input type="button" value="Edit"/>
Makkonen Eino	A	3	A5	CLPILNOF86	<input type="button" value="Edit"/>
Heiskanen Lauri	A	3	A6	J7BLDWECZV	<input type="button" value="Edit"/>

Kuva 17 Asukastietojen hallinta

Asukastietojen hallintataulukossa on käytetty jQuery -kirjastosta löytyvää Tabledit -liitännäistä, joka on yhteensopiva Bootstrap:n kanssa. Tabledit:n avulla taulukon tietoja voidaan muokata nopeasti päivittämättä koko sivustoa. Kuvassa 18 on esimerkki Tabledit -liitännäisestä, kuvankaappaus otettu jQuery-Tabledit:n Github sivustolta (Marques, 2017).

Table JavaScript

```

$('#example1').Tabledit({
  url: 'example.php',
  columns: {
    identifier: [0, 'id'],
    editable: [[1, 'nickname'], [2, 'firstname'], [3, 'lastname']]
  }
});

```

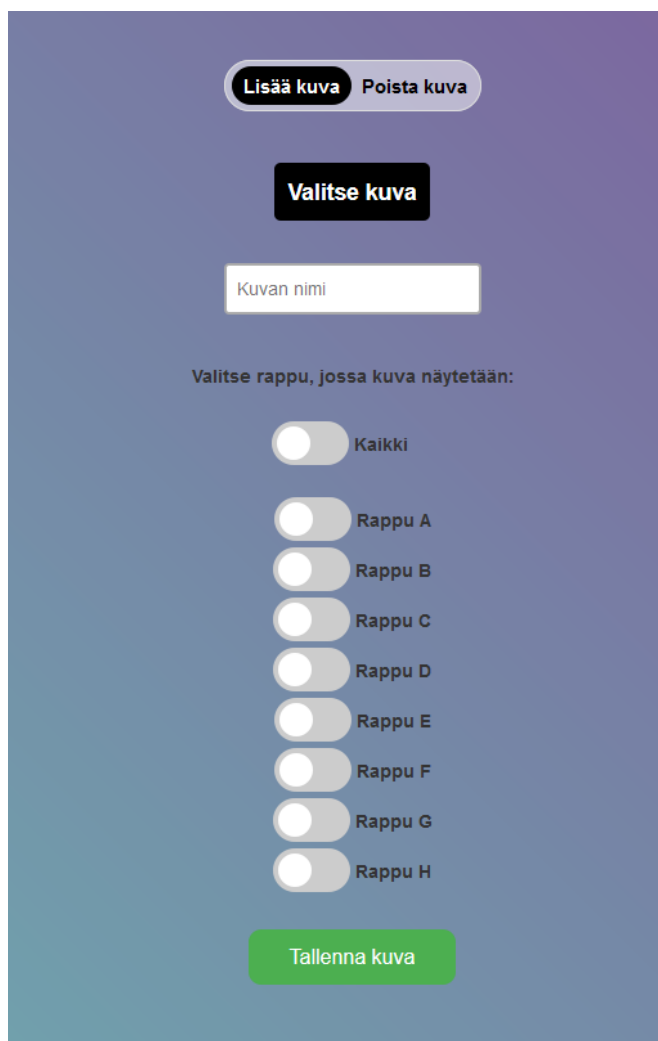
#	NICKNAME	FIRSTNAME	LASTNAME	
1	markcell	Celso	Marques	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	dotz	Márcio	Quental	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	zikospeed	António	Figueiredo	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Kuva 18 Tabledit -esimerkki

4.5.2 Ilmoitusten hallinta

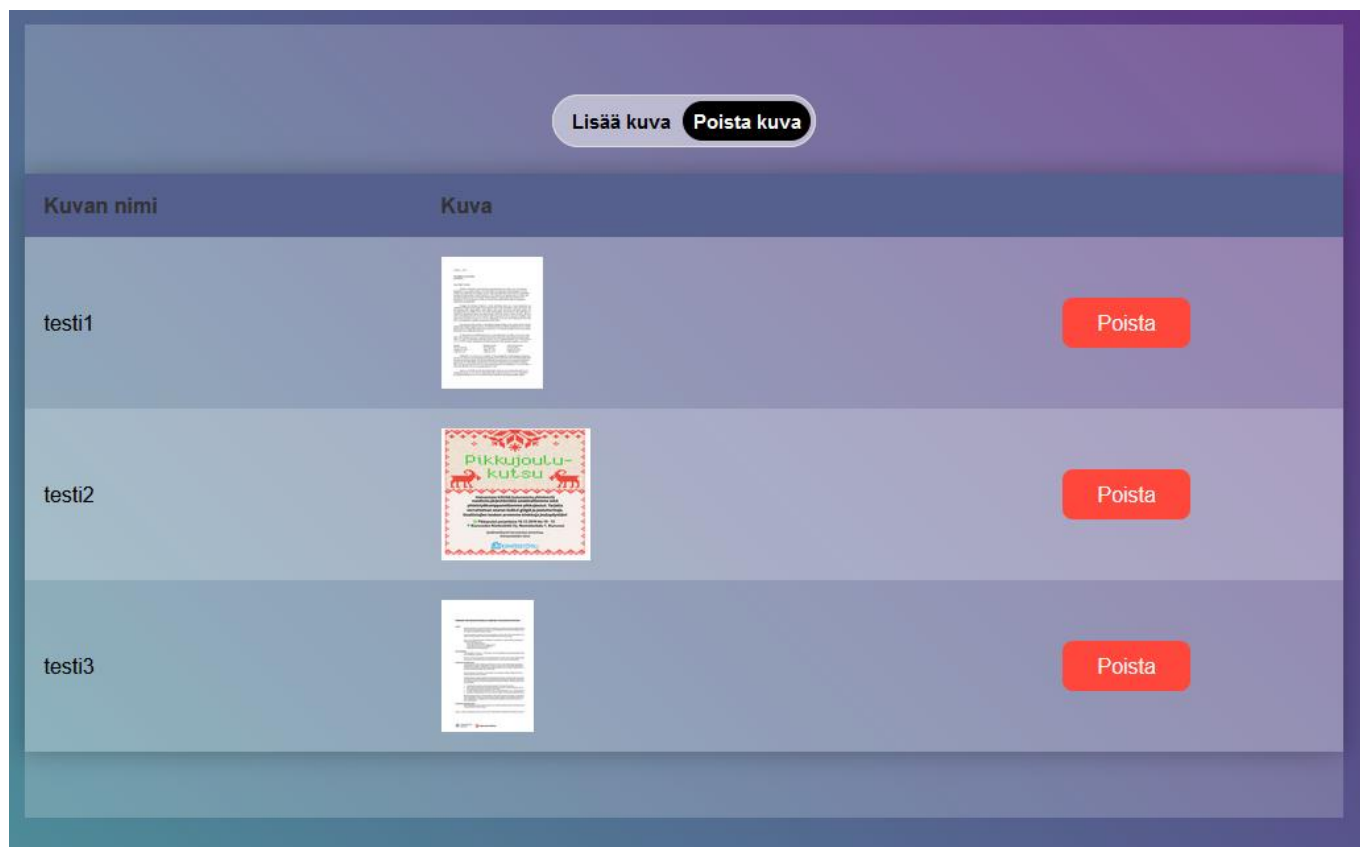
Ilmoitusten lisääminen toteutettiin kuva -muodossa suoraan palvelimelle, eikä tässä tilanteessa kuvien tiedot siis tallennu tietokantaan. Ilmoitukset -näköymästä löytyy kaksi välilehteä, kuvien lisääminen ja poistaminen. Kuvaa lisättäessä annetaan sille nimi ja valitaan rappukäytävät, joiden ilmoitustaululle ilmoitus halutaan näkyviin. Ilmoitusten poistaminen tapahtuu Poista kuva -välilehdeltä, josta nähdään kuvan nimi ja itse kuva voidaan avata uuteen selaimen välilehteen täysikokoisena klikkaamalla pientä kuvaa. Kuvan poistaminen pysyvästi palvelimelta tapahtuu Poista -painikkeella.

Kuvassa 19 näkymän ilmoituksen lisäämisestä, johon tulossa vielä kuvasta poiketen ilmoituksen ajan lisäys, kuinka kauan ilmoitus näkyy ilmoitustaululla. Kuvassa 20 nähdään ilmoitusten poistaminen -välilehti.



The screenshot shows a form for adding a new advertisement. At the top, there are two buttons: 'Lisää kuva' (Add image) and 'Poista kuva' (Remove image). Below them is a 'Valitse kuva' (Select image) button. A text input field is labeled 'Kuvan nimi' (Image name). Underneath, the instruction 'Valitse rappu, jossa kuva näytetään:' (Select the room where the image will be displayed:) is followed by a list of eight rooms, each with a toggle switch: 'Kaikki' (All), 'Rappu A', 'Rappu B', 'Rappu C', 'Rappu D', 'Rappu E', 'Rappu F', 'Rappu G', and 'Rappu H'. At the bottom of the form is a green 'Tallenna kuva' (Save image) button.

Kuva 19 Uuden ilmoituksen lisääminen



Kuva 20 Ilmoitusten poistaminen

Kuvien hallinnassa on käytetty AJAX -tekniikkaa, jolla dataa vaihdetaan palvelimen kanssa siten, ettei koko sivustoa tarvitsisi ladata uudelleen.

4.5.3 Varausten hallinta

Varausten hallintaan olisi löytynyt netistä monenlaisia valmiita kalenteritoteutuksia, kuten JavaScript:n Fullcalendar, mutta tämä päätettiin toteuttaa itse alusta alkaen. Tällä tavalla varausten kalenterinäköymästä saatiin juuri sellainen kuin haluttiin ja välttyttiin eri tekniikoiden integroimisen tuomista haasteista. Tällä tavoin kalenterinäköymästä tuli myös yksinkertainen ja helpokäyttöinen.

Varauksia voidaan tarkastella kalenteri- tai luettelonäköymästä. Kalenterinäköymässä nähdään kerrallaan yhden viikon varaukset ja navigointi -nuolella voidaan tarkastella seuraavan viikon varauksia. Kuvan 22 kalenterinäköymän oikeasta yläreunasta voidaan vaihtaa tarkasteltavaa kohdetta ja varauksien poisto tapahtuu painamalla kalenterissa näkyvästä nimestä.

Kalenteri **Luettelo**

Kohde	PVM	Aika	Nimi
Pyykkitupa1	15/02/2020	09:00	Pekkarinen Pauliina
Pyykkitupa2	13/02/2020	12:00	Toivonen Anni
Pyykkitupa2	13/02/2020	11:00	Toivonen Anni
Sauna1	15/02/2020	11:00	Makkonen Eino
Pyykkitupa1	23/02/2020	16:00	Toivonen Anni
Pyykkitupa1	28/02/2020	12:00	Toivonen Anni
Pyykkitupa1	22/04/2020	12:00	Toivonen Anni
Pyykkitupa1	25/04/2020	17:00	Makkonen Eino
Pyykkitupa1	26/04/2020	19:00	Tuomainen Mauri

Kuva 22 Varauksien luettelo -näkyvä

Kalenteri **Luettelo**

Pyykkitupa1

	21.04.2020	22.04.2020	23.04.2020	24.04.2020	25.04.2020	26.04.2020
07:00						
08:00						
09:00						
10:00						
11:00						
12:00		Toivonen Anni				
13:00						
14:00						
15:00						
16:00						
17:00					Poista	
18:00						
19:00						Tuomainen Mauri

Kuva 21 Varauksien kalenteri -näkyvä

Varausten kalenterin toiminta perustuu pääosin AJAX:n XMLHttpRequest -tekniikkaan, jolla varaustietoja tuodaan tietokannasta lataamatta koko sivua uudelleen. Tällä tekniikalla data saadaan palvelimelta vasta sivun latauksen jälkeen. Kuvassa 23 nähdään lyhyitä otoksia sivuston koodista, jossa käytetty XMLHttpRequest -tekniikkaa.

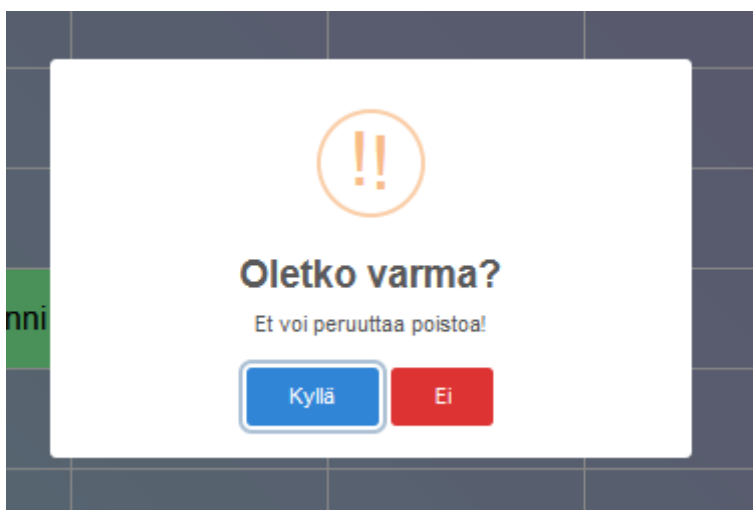
```
if (window.XMLHttpRequest) {  
    xmlhttp = new XMLHttpRequest();  
} else {  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}  
  
xmlhttp.open("GE  
xmlhttp.send();
```

Kuva 23 Esimerkki XMLHttpRequest -koodista

4.5.4 Ponnahdusikkunat

Sovelluksen erilaiset varoitukset ja toimintojen loppuun viemisen varmistamiset toteutettiin ponnahdusikkunoilla. JavaScript:n perinteisten ponnahdusikkunoiden sijaan jokaisen sovelluksen osalualueilla käytettiin GitHub -käyttäjän Limon Monten kehittämää Sweetalert2 -ponnahdusikkunaa. Se on laajasti muokattavissa, tavallista ponnahdusikkunaa näyttävämpi versio.

Kuvassa 24 on varauksen poistamisen varmistamiseen käytetty Sweetalert -ponnahdusikkuna.



Kuva 24 Esimerkki Sweetalert -ponnahdusikkunasta

5 POHDINTAA INFONÄYTÖISTÄ

5.1 Vaatimukset ja hintataso

Rappukäytäviin sijoitettavat näytöt eivät tule olemaan tavallisia sähköisiä näyttöjä, kuten televisiot tai tietokoneen näytöt. Tähän tarkoitukseen sopivan näytön pitäisi olla infonäyttö, joka vastaa siihen kohdistuviin haasteisiin, kuten kestävyyteen näytön ollessa päällä jatkuvasti, jopa 24/7. Infonäyttöjen hinnoittelu on yleensä korkeampaa kuin tavallisissa näytöissä, mutta kestävyys ja ominaisuudet ovat parempia kyseiseen tarkoitukseen. Tässä projektissa käytettäviltä laitteilta ei kuitenkaan vaadita kosketusominaisuutta, jonka vuoksi näyttöjen hinnat eivät nouse mahdottoman korkealle. Jos näytönhankinta budjetti on erityisen matala, voidaan harkita tähän ilmoitustaulu käyttöön myös tavallisia näyttöjä, mutta tällaisessa tilanteessa joudutaan tekemään kompromisseja käytön suhteen. Näyttöjä voidaan esimerkiksi sammuttaa silloin, kun ne eivät ole aktiivisessa käytössä, kuten öisin.

Infokäyttöön tarkoitettujen näyttöjen hintatasoa lähdettiin tarkastelemaan pikaisilla Google -hauilla, joista selvisi kuluttajahintatason alkavan noin 200 eurosta ylöspäin. Todellinen hintataso määräytyy, kun päätetään käytettävän näytön kokoluokka ja mitä ominaisuuksia siltä vaaditaan. Joissakin näytöissä on valmiina käyttöjärjestelmiä, joilla saadaan esimerkiksi nettiyhteyden avulla aukaistua näytölle web-sivusto, joka olisi tässä projektissa oivallinen toteutusvaihtoehto laitteiston puolelta.

6 POHDINTA JA JATKOKEHITYS

6.1 Pohdinta

Projektin tavoitteena oli luoda Asunto-osakeyhtiö Kuopion Hiihtäjätie 15 rappukäytävälle tuleville digitaalisiin näytöille helppokäyttöinen ja toimiva ilmoitustaulu -sovelluskokonaisuus.

Mielestäni tavoitteessa onnistuttiin hyvin, sovelluksesta saatiin toimiva kokonaisuus, jonka opinnäytetyösopimuksessa laaditut vaatimukset saatiin myös toteutettua. Aikataulullisesti projekti saatiin valmiiksi hieman etuajassa, jolloin sovelluksen viimeistelyyn jää vielä hyvin aikaa, ennenkuin sitä aletaan viemään taloyhtiön rappukäytäviin. Mielestäni varsin hyvin onnistuttiin siinä, että sovelluksesta saatiin helppokäyttöinen ja yksinkertainen.

Sovelluksesta kuitenkin vielä puuttuu muutamia viimeisteltäviä asioita, kuten hallintasovelluksen eli taloyhtiön käyttäjien hallinta, johon kuuluu käyttäjien lisääminen, poistaminen ja salasanojen vaihto. Nämä ehditään saada valmiiksi, ennenkuin näyttöjen hankinta ja sovelluksen integroiminen aloitetaan.

Projektin aikana mahdolliseksi haasteeksi ilmeni kevään 2020 aikana maailmalle levinnyt COVID-19- eli koronavirus -pandemia. Koronaviruksen johdosta hallitus antoi erilaisia ohjeita ja linjauksia, joissa esimerkiksi henkilöiden välisiä tapaamisia ja lähikontakteja pyydettiin välttämään. Tämä ei itseni kohdallani kuitenkaan aiheuttanut haasteita projektin etenemisen kannalta, koska pääosin sovellusta tehtiin kotoa käsin tietokoneelta. Projektin loppuunvieminen voi kuitenkin viivästyä, mikäli erilaiset henkilöiden ja lähikontaktien tapaamiskiellot tiukentuvat.

Haastavimmaksi asiaksi projektin aikana koin MySQL -lausekkeiden toteuttamisen varsinkin hallintasovelluksen varauskalenterin kohdalla. Yksinkertaiset MySQL -lausekkeet eivät yleensä tuota ollenkaan vaikeuksia, mutta tämän kohdalla jouduttiin tekemään useampi pidempi ja monimutkaisempi lauseke, jotta kalenterista saatiin toimiva.

6.2 Jatkokehitysideoita

Sovellukseen voitaisiin kehittää vielä ainakin toiminto, jolla asukkaat voivat itse luoda ilmoituksia, joita julkaistaisiin ilmoitustaulu -näytöllä taloyhtiön hyväksynnän jälkeen. Näin asukkaat voisivat ilmoittaa asioista, jotka koskisivat muitakin asukkaita tai esimerkiksi ilmoittaa pitämistään tupaantulojaisista.

Asukkaille voitaisiin myös tehdä jonkinlainen chat -tyylinen näkymä, jossa asukkaat voisivat keskenään vaihdella kuulumisiaan ja mielipiteitään erilaisista asioista tai varoittaa esimerkiksi pyörävarkaista tai ilmoittaa rappukäytävien ympäristöstä löytyneistä tavaroista.

Ilmoitustaulu -näkymlle voitaisiin myös lisätä erilaista ajankohtaista sisältöä, kuten säätiedotukset ja uutisnäkymä, jossa näytettäisiin tiivistelmiä päivän tärkeimmistä uutisista.

7 KUVAT

Kuva 1. Sovelluksessa käytetty teknologia.....	8
Kuva 2 Esimerkki HTML -pohjasta, alapuolella selainnäkyä.	10
Kuva 3 Esimerkki CSS -tyylittelystä, kuvassa oikealla selainnäkyä.	11
Kuva 4 Esimerkki PHP-koodista.	15
Kuva 5 Suunnitteluvaiheen ensimmäinen näyttökuva	18
Kuva 6 Suunnitteluvaiheen näyttökuva	20
Kuva 7 phpMyAdmin -hallintatyökalu	22
Kuva 8 Tietokannan rakenne.....	23
Kuva 9 Ilmoitustaulu -näkyä	24
Kuva 10 Asukkaan varaustietojen välilehdeksi.....	25
Kuva 11 Responsiivisuuden vinkkejä	26
Kuva 12 Varauksen ajankohdan valitseminen	26
Kuva 13 Varauksen onnistuminen	27
Kuva 14 Asukkaan varausten hallinta	27
Kuva 15 Hallintasovelluksen kirjautumisnäkyä	28
Kuva 16 Hallintasovelluksen etusivunäkyä	29
Kuva 17 Asukastietojen hallinta	30
Kuva 18 Tabledit -esimerkki	30
Kuva 19 Uuden ilmoituksen lisääminen	31
Kuva 20 Ilmoitusten poistaminen.....	32
Kuva 22 Varauksien kalenteri -näkyä.....	33
Kuva 21 Varauksien luettelo -näkyä	33
Kuva 23 Esimerkki XMLHttpRequest -koodista	34
Kuva 24 Esimerkki Sweetalert -ponnahdusikkunasta	34

8 LÄHTEET

- Bourdon, R. *WampServer*. Haettu 10. 3. 2020 osoitteesta <http://www.wampserver.com/en/>
- Delgado, T. H. (25. 1. 2019). *Web Design akus.net*. Haettu 10. 2. 2020 osoitteesta <https://disenowebakus.net/en/html-history>
- Dwika, H. (4. 3. 2020). *Hostinger*. Haettu 22. 3. 2020 osoitteesta <https://www.hostinger.com/tutorials/what-is-mysql>
- Forsström, M. (1. 10. 2015). *Fraktio*. Haettu 16. 2. 2020 osoitteesta <https://fraktio.fi/blogi/javascriptin-lyhyt-historia/>
- Grant, A. (6. 12. 2019). *Make Use Of*. Haettu 16. 2. 2020 osoitteesta <https://www.makeuseof.com/tag/what-is-javascript/>
- Guo, D. (13. 8. 2019). *Logrocket*. Haettu 19. 2. 2020 osoitteesta <https://blog.logrocket.com/the-history-and-legacy-of-jquery/>
- Jaiswal, S. (2018). *Javatpoint*. Haettu 22. 3. 2020 osoitteesta <https://www.javatpoint.com/mysql-history>
- Kinsta*. (2. 3. 2020). Haettu 18. 3. 2020 osoitteesta <https://kinsta.com/knowledgebase/what-is-github/>
- Marques, C. (2017). *Github*. Haettu 28. 3. 2020 osoitteesta <https://markcell.github.io/jquery-tabledit/#examples>
- MDN-contributors. (16. 3. 2020). *MDN web docs*. Haettu 20. 3. 2020 osoitteesta <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Moraes, F. (1. 1. 2020). *HTML.COM*. Haettu 10. 2. 2020 osoitteesta <https://html.com/>
- Morris, S. (2020). *Skillcrush*. Haettu 21. 3. 2020 osoitteesta <https://skillcrush.com/blog/php/>
- Morris, S. (2020). *SkillsCrush*. Haettu 18. 2. 2020 osoitteesta <https://skillcrush.com/blog/what-is-jquery-used-for/#more>
- Nawaz, S. (4. 1. 2019). *Cloudways*. Haettu 21. 3. 2020 osoitteesta <https://www.cloudways.com/blog/php-version-history/>
- PHP.net. (28. 11. 2019). *PHP*. Haettu 28. 3. 2020 osoitteesta <https://www.php.net/archive/2019.php#2019-11-28-1>
- SimpleCSS*. Haettu 15. 2. 2020 osoitteesta <https://simplecss.eu/css-history-brief-overview.html>
- w3school*. Haettu 21. 3. 2020 osoitteesta <https://www.w3schools.in/bootstrap4/intro/>
- Yogihosting*. (1. 3. 2020). Haettu 20. 3. 2020 osoitteesta <https://www.yogihosting.com/what-is-bootstrap/>