

TALOSUUNNITTELUOHJELMAN KÄYTTÖLIITTYMÄ



Ammattikorkeakoulututkinnon opinnäytetyö

Riihimäki, Tietotekniikka

Kevät, 2017

Nicholas Nurminen

Tietotekniikka
Riihimäki

Tekijä	Nicholas William Nurminen	Vuosi 2017
Työn nimi	Talosuunnitteluohjelman käyttöliittymä	
Työn ohjaaja/t	Petri Kuittinen	

TIIVISTELMÄ

Työn idea tuli Masterhouse Oy -nimiseltä piharakennusyritykseltä. Se (yri-ty)s) tarvitsisi selaimeen sijoitettavan ohjelman, jolla asiakas pystyisi suunnittelemaan helposti omiin tarpeisiinsa soveltuvan piharakennuksen erilaisista elementtiosista. Tästä suunnitelmasta asiakkaan oli tarkoitus pystyä lähettämään tiedot yritykselle tarjouspyyntöä varten.

Masterhouse Oy:n projektin laajuus ei olisi soveltunut opinnäytetyöhön suunniteltuun työmäärään, joten ohjelmasta päätettiin tehdä opinnäyte-työn puitteissa vain toimiva prototyyppi. Tämän prototyypin pohjalta saa-tettaisiin rakentaa lopullinen versio opinnäytetyön jälkeen tai vaihtoehto-isesti käyttää jotain muuta ratkaisua.

Käyttöliittymä toteutettiin hyödyntämällä yleisimpiä web-tekniikoita, jot-ka ovat merkkaukieli HTML5, ohjelmointikieli JavaScript ja kuvauskieli CSS. Opinnäytetyössä käydään läpi edellä mainittujen web-tekniikoiden vahvuuksia sekä heikkouksia ja tutustutaan jQueryyn ja Bootstrappiin.

Käyttöliittymästä saatiin valmiiksi toimiva prototyyppi. Tulos ei välttämät-tä ollut niin nätti kuin olisi voinut toivoa, mutta käyttöliittymässä on kaik-ki tarvittavat ominaisuudet, jotka siihen suunniteltiin. Ainoa ero on se, et-tä suunnitelma ladataan prototyypissä omalle koneelle sen sijaan, että se lähetettäisi yritykselle.

Lopputuloksena oli loppujen lopuksi onnistunut, sillä ohjelma tekee sen mitä sen täytyykin. JQueryn ja Bootstrapin käyttäminen olisi nopeuttanut to-teutusprosessia ja parantanut lopputulosta, mutta niihin tutustuminen tapahtui liian myöhäisessä vaiheessa. Tästä syystä opinnäytetyön jälkeen teen ohjelman yritykselle uudestaan paremmalla osaamisella.

Avainsanat HTML5, JavaScript, CSS, Suunnitteluohjelma, Käyttöliittymä

Sivut 26 sivua

Information Technology
Riihimäki

Author	Nicholas Nurminen	Year 2017
Subject	User interface for house design program	
Supervisors	Petri Kuittinen	

ABSTRACT

The idea for the thesis came from Masterhouse Oy. They wanted an application that their customers could use to design small warehouses, which are made of different kinds of elements. This warehouse design could afterwards be sent to Masterhouse Oy for a possible deal to be made.

The scope of this project was excessively big for bachelor's thesis limit, so it was decided that only a functional prototype of the program would be made. This prototype could be continued after the bachelor's thesis or a new version of the program would be made.

User interface was made with using the most universal and used web-technologies. These are HTML5 markup language, JavaScript programming language and CSS style sheet language. The bachelor's thesis also examines their positive and negative qualities. jQuery and Bootstrap will be examined, even though they weren't used in the prototype.

A complete prototype of the user interface was achieved. The outcome was not as good looking as it could have been, but it included almost all the necessary features that were planned. The only thing that differed in prototype was the sending of design. Instead of sending it to the company, it was downloaded to the user's computer.

The result was quite successful, even though it had its flaws. At least it did what it was supposed to do. Earlier interaction and reading about jQuery and Bootstrap would have affected the result positively and made the process faster, but I learned about them too late. For this reason, I will remake the program with better skills and knowledge.

Keywords HTML5, JavaScript, CSS, Design Program, User Interface

Pages 26 pages

SISÄLLYS

1	JOHDANTO.....	1
2	SUUNNITTELU.....	1
2.1	Web-tekniikat.....	1
2.1.1	HTML.....	1
2.1.2	JavaScript.....	2
2.1.3	CSS.....	4
2.2	Avoimen lähdekoodin kirjastot.....	5
2.2.1	JQuery.....	5
2.2.2	Bootstrap.....	6
2.3	Ominaisuuksien kartoitus.....	6
2.3.1	Opinnäytetyöhön kuuluvat ominaisuudet.....	6
2.3.2	Opinnäytetyön ulkopuolelle jäävät ominaisuudet.....	7
2.4	Työkalut.....	7
2.4.1	Tekstieditori.....	7
2.4.2	Graafiset työkalut.....	8
2.5	Ohjelmien ulkoasun suunnittelu.....	8
2.5.1	Ensimmäinen suunnitelma.....	8
2.5.2	Toinen suunnitelma.....	9
2.5.3	Kolmas suunnitelma.....	10
3	TOTEUTUS.....	11
3.1	Drag and copy.....	12
3.2	Ruudukko.....	13
3.3	Prototyypipalat.....	15
3.4	Poistaminen.....	16
3.5	Informaatio kentät.....	16
3.6	Tallentaminen.....	17
3.7	Virheiden korjaus ja hienosäätö.....	18
3.7.1	Valikon kopioituminen.....	19
3.7.2	Valikon asettelu.....	19
3.7.3	Elementtiosien kuvien linkitys.....	21
4	YHTEENVETO.....	22
4.1	Lopputulos.....	22
4.1.1	JQuery.....	22
4.1.2	Bootstrap.....	22
4.2	Arviointi.....	22
	LÄHTEET.....	24

1 JOHDANTO

Työ tehtiin Masterhouse Oy -nimiselle yritykselle. Työn tarkoituksena on suunnitella käyttöliittymä pienrakennusten suunnitteluohjelmalle ja luoda prototyyppi kyseisestä ohjelmasta. Ohjelma tulisi olla sijoitettavissa internet-selaimiin, jossa asiakas kykenisi itse vaivatta suunnittelemaan halutun kaltaisen rakennuksen pohjaluonnoksen ja lähettää tarjouspyynnön yritykselle tarjouslaskentaa varten. Työtä ei ole tarkoitus viedä alusta loppuun saakka, vaan tehdä ohjelmasta toimiva prototyyppi, josta olisi mahdollista jatkaa myöhemmin valmiiseen versioon.

Asiakasyrityksellä itsellään ei ole kokemusta verkkosivujen tai sovellusten tekemisestä, joten toteutustapa jäi vapaasti valittavaksi, kunhan se soveltuu selaimessa käytettäväksi.

Työn tarkoituksena minulle opiskelijana on kehittää omia käyttöliittymäsuunnittelu taitoja ja saada käytännön kokemusta asiasta. Sivussa tulee myös virkistettyä ja hiottua ohjelmoinnin taitoja tehdessä prototyyppiä tai pohjaa käyttöliittymälle. Työn painopisteenä tulee kuitenkin olemaan erilaisten käyttöliittymien suunnittelemisen harjoittelu ja ohjelmasta kärkeän prototyypin luominen.

2 SUUNNITTELU

Työn tilaajalla ei ole tällä hetkellä käytössään vastaavanlaista ohjelmaa, eikä vastaavanlaisia ohjelmia tällä hetkellä ole tietääkseni suunniteltu yritysten asiakkaiden käytettäväksi. Nykyiset versiot suunnitteluohjelmista on tarkoitettu ammattilaisten käyttöön ja tästä syystä liian monimutkaisia ja vaikeakäyttöisiä yritysten asiakkaiden käytettäväksi. Työn toteutus vaatii joitakin työkaluja, koodikieliä sekä editoreita.

2.1 Web-tekniikat

Web-tekniikat ovat koodikieliä, jotka on tarkoitettu verkkosivujen ja sovelluksien tekemiseen. Yleisimpiä Web-tekniikoita ovat avoimesti standardoitu kuvauskieli HTML, dynaaminen komentosarjakieli JavaScript ja tyyliohjeiden laji CSS.

2.1.1 HTML

HTML eli Hyper Text Markup Language on avoimesti standardoitu kuvauskieli. Uusin standardi tästä kuvauskielestä on HTML5, joka sai W3C-

suosituksen 2014 lokakuussa (W3Schools HTML5, n.d.). Monet webiselaimet ovat tukeneet HTML5:n uusia ominaisuuksia ja lisäksi monet verkkosivut ovat siirtyneet HTML5:n käyttämiseen.

HTML-verkkosivu koostuu erilaisista HTML-elementeistä. Elementti koostuu alku-tägistä, jonka jälkeen tulee haluttu sisältö, minkä jälkeen taas suljetaan elementti. Alla olevassa koodissa on esimerkki, jossa luodaan title-tägi, jonka sisään laitetaan haluttu sisältö. Se on tässä tapauksessa tekstiä ja suljetaan jälkepäin uudella tågillä.

```
<title>Suunnittele Oma Rakennuksesi</title>
```

HTML5:n ominaisuuksia ovat mm. <canvas>-, <video>- ja <audio> -elementit. HTML5 mahdollistaa myös SVG-muotoisen (Scaleable Vector Graphics) sisällön näyttämisen käyttämättä <object> -tägiä. HTML5:ssä on myös merkityksiä rikastuttavia ominaisuuksia, kuten <section>-, <article>-, <header>- ja <nav>-täggit ja uusia tågien määrittelemiseksi ja selkeyttämiseksi tehtyjä attribuutteja. (Wikipedia HTML5, n.d.)

HTML5-verkkosivuilla on edeltävää HTML4-versiota paljon lyhyempi doctype. HTML5 doctype on myös paljon yksinkertaisempi verrattuna edeltävään versioon. Alla vertailtavissa ensin strict-tyyppinen HTML4 doctype ja HTML5 doctype.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE HTML>
```

HTML Canvas on hämmästyttävä piirustustekniikka, joka sisältyy kaikkiin nykyaikaisiin web-selaimiin. Sillä voi piirtää kuvioita, manipuloida kuvia, rakentaa pelejä ja animoida käytännössä mitä tahansa, voit jopa luoda mobiilisovelluksia. (Marinacci 2012.)

2.1.2 JavaScript

JavaScript on erittäin yleinen ohjelmointikieli, jolla voidaan luoda verkkosivuun erinäisiä dynaamisia ominaisuuksia. Sitä voidaan mm. käyttää HTML-sivuja luodessa, jos websivuille halutaan lisätä käyttäjäpuolen interaktiivisuutta tai lisätä jotain ominaisuuksia web-sivuille. (Wikipedia JavaScript, n.d.)

JavaScriptin esiintyvyyden takia internetissä ja muualla, JavaScriptistä on tullut vaikeasti vältettävä. Tämä ei kuitenkaan tarkoita sitä, että siitä ei pidettäisi. Vaikka JavaScriptissä onkin paljon oikkuja käytettäessä, se on käypä ohjelmointikieli, joka tekee käyttäjästäan tuotteliaan ja sillä voi olla myös hauska ohjelmoida. (Rauschmayer, 2014.)

Nimestään huolimatta JavaScript on täysin eri ohjelmointikieli kuin Java, pois lukien syntaksin yhteneväisyydet. Tämä johtuu JavaScriptin julkaisuajasta, jolloin Java-ohjelmointikieli oli suosiossa, mikä johti etuliitteen lisäämiseen markkinointitarkoituksessa. Tästä huolimatta JavaScript on itsenäinen ohjelmointikieli ja yleisessä käytössä tänä päivänäkin. (Haverbeke, 2014.)

JavaScriptistä on ollut olemassa useita eri versioita. ECMAScript-versio 3 oli laajasti tuettu versio sillä hetkellä kun JavaScript nousi yleisimmäksi ohjelmointikieleksi vuosien 2000 ja 2010 välillä. Se kävi läpi useita versioita, mutta vuonna 2009 tullut viides versio on nykyään kaikkien selainten tukema. Kuudes versio on tällä hetkellä tekeillä ja jotkin selaimet ovat alkaneet tukemaan sen ominaisuuksia. (Haverbeke, 2014.)

Tällä hetkellä suurin osa nykyaikaisista websivuista käyttää JavaScriptiä, sekä kaikki nykyaikaiset selaimet, peli- sekä mobiililaitteet tukevat sitä, mikä tekee JavaScriptistä yhden yleisimmistä ohjelmointikielistä tällä hetkellä. Myös tietokannat, kuten MongoDB ja CouchDB käyttävät JavaScriptiä, kuten myös serveripuolen ohjelmointi, kuten Node.js (Haverbeke, 2014.).

JavaScriptillä pystyy muokkaamaan HTML-sisältöä ja ominaisuuksia. Sillä pystyy myös vaihtamaan CSS-tyyliä, piilottamaan HTML-elementtejä, sekä palauttamaan näkyville piilotettuja HTML-elementtejä. Nämä ovat esimerkkejä siitä, mitä JavaScriptillä pystyy tekemään. Alla olevassa HTML esimerkissä nappulan onclick-toimintoon on lisätty JavaScript-toiminto, joka muuttaa "JavaScript voi muuttaa HTML sisältöä." tekstin "Hello World!" tekstiksi.

```
<!DOCTYPE html>
<html>
  <body>

  <p id="testi">JavaScript voi muuttaa HTML si-
sältöä.</p>

  <button type="button" on-
click='document.getElementById("testi").innerH
TML = "Hello World!'">Muutos
  </button>

  </body>
</html>
```

2.1.3 CSS

CSS eli Cascading Style Sheets, on verkkosivuja varten tehty tyylikieli. Niiden avulla voidaan ohjata erinäisten HTML-elementtien ulkonäöllisiä seikkoja ja sommittelua. Vaihtoehtoisesti HTML-elementeille voi lisätä myös id- tai luokka-muuttujan. Näin on helpompi erotella kohdat, joilla on sama HTML-elementti, mutta vaativat hieman tai kokonaan erilaisen muotoilun. (Wikipedia CSS, n.d.)

Alla olevassa esimerkissä kyseessä olevan sivun bodyn taustaksi on valittu black, eli musta. Värin voi valita myös värikoodilla (musta on #000000). Toisena tiedostossa on h1, joka tarkoittaa ensimmäisen asteen otsikkoa, jonka väriksi on valittu punainen ja se on sijoitettu keskelle näyttöä. Viimeisenä esimerkissä on p, joka tarkoittaa tekstiä, tulee englanninkielisestä sanasta paragraph. Tässä on asetettu tekstin fontiksi verdana ja kooksi 20 pikseliä.

```
body {  
    background-color: black;  
}  
  
h1 {  
    color: red;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

CSS-tiedosto sisältää paljon sääntöjä, joilla sivusto saadaan helposti näyttämään yhdenmukaiselta ilman tarvetta toistaa samoja visuaalisia ominaisuuksia muokkaavia koodeja. CSS ei tuo websivulle lisää sisältöä, vaan sen tarkoitus on helpottaa tiedon esittämistä. CSS-tiedosto on käytännössä tekstitiedosto, jonka päätteenä on ".css". Csstiedostoon viitataan html-tiedostossa, jolloin siellä olevat tágit tulevat html-tiedostoon käytettäväksi.

CSS-tyylitiedoston käyttämisessä on monia etuja. Sen avulla on helppo muokata sivustoa jälkeenpäin. Fontit, värit, sommittelu ja koot on valmiiksi määriteltä ja pätee asetettuihin HTML-elementteihin automaattisesti. Etuna on myös, että tyylin muokkaaminen tai kokonainen vaihtaminen on suhteellisen helppoa, sillä tyyli-tiedostoa tarvitsee muokata vain yhdestä paikasta. Verkkosivun tiedot eivät myöskään katoa yltiömääräisen visuaalisen muotoilun sekaan, vaan kaikki pysyy hyvin järjestyksessä ja helposti muokattavissa.

2.2 Avoimen lähdekoodin kirjastot

JQuery ja Bootstrap ovat avoimen lähdekoodin kirjastoja, jotka soveltuvat kaikille selaimille. Avoimen lähdekoodin kirjasto on sivusto, josta voi ladata erilaisia ominaisuuksia, jotka vaikuttavat ohjelmointiin. Tässä tapauksessa ne vaikuttavat websivujen ohjelmointiin siten, että ne antavat valmiita vaihtoehtoja, joita voi käyttää websivuja tehdessä. Bootstrap vaatii toimiakseen myös jQueryn.

Näiden kirjastojen hyötyinä on toistamisen välttäminen projektien välillä, helposti luotava mukautuminen esimerkiksi erikokoisille ruuduille, nopeasti prototyyppien luominen ja selainten välinen yhteensopivuus. Tämän lisäksi useampi kehittäjä saa samanlaisen designin aikaiseksi helposti, sillä tyyliäännöt tulevat valmiista kirjastosta. (Rascia, 2015.)

2.2.1 JQuery

JQuery on erittäin suosittu JavaScript-koodikehys eli kirjasto. Se on tarkoitettu kaikille selaimille ja sen käyttäminen on täysin ilmaista. Siitä on tehty myös mahdollisimman yksinkertainen, jotta sitä olisi helppo ymmärtää ja käyttää. (jQuery, n.d.)

JQueryn käyttäminen toimii siten, että ladataan kopio, joka voidaan linkittää www-sivuun. Suurin osa jQuerystä löytyvistä ominaisuuksista liittyy joko HTML-dokumenttien manipulointiin, tapahtumien hallintaan, animointiin tai Ajax-sovelluksiin. (jQuery, n.d.)

JQuery-kirjasto tekee helpommaksi manipuloida HTML-sivua itse selaimessa. Se antaa myös työkaluja, jotka helpottavat käyttäjän vuorovaikuttamista sivun kanssa, tekevät animaatioiden luomisesta sivulle helpompaa ja antavat käyttäjän kommunikoida serverin kanssa päivittämättä sivua. (Murphey, 2012.)

JQueryn käyttämisestä on oiva esimerkki w3schools sivuilla. Esimerkissä käytetään hide-toimintoa piilottamaan tekstiä, eli "p" html-elementeissä olevaa sisältöä, mikäli sitä klikkaa hiiren vasemmalla painikkeella. (W3schools, n.d.)

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function() {
    $("p").click(function() {
        $(this).hide();
    });
});
</script>
```

Script elementin alkuun on laitettu etsimiskomento, joka hakee tarvittavat JavaScript-toiminnot Googlen isännöimästä kirjasto-palvelusta. Tämän jälkeen alle lisätään toiminto, joka aiheuttaa sen, että kun ”p” html-elementtiä klikkaa, se katoaa.

2.2.2 Bootstrap

Bootstrap on ilmainen web-kehys, jolla on avoimen lähdekoodin kirjasto, joka on suunnattu lähinnä websivujen muotoiluun ja webohjelmien toimintoihin. Se sisältää HTML- ja CSS-pohjaisia muotoilumalleja kaavioille, nappuloille, websivuilla navigoinnille ja muita käyttöliittymän komponentteja. Tämän lisäksi <bootstrapista löytyy myös vapaavalintaisia JavaScript-laajennuksia. Bootstrap keskittyy vain käyttöliittymäsuunnitteluun tarkoitettuihin ominaisuuksiin. (Bootstrap, n.d.)

Bootstrapin hyviä puolia ovat helppokäyttöisyys, mukautuvat ominaisuudet ja selainten kanssa yhteensopivuus. Kuka tahansa, jolla on perusymmärrys HTML:stä ja CSS:stä voi alkaa käyttää Bootstrappia. Ominaisuudet mukautuvat hyvin puhelimesta tabletin kautta pöytäkoneisiin. Bootstrap 3, joka on tällä hetkellä uusin versio Bootstrapista, tukee erityisesti mobiililaitteita. (W3schools, n.d.)

2.3 Ominaisuuksien kartoitus

Ohjelmalla tulee olla joitakin ominaisuuksia minimissään, jotta ohjelma toimii ja se pystyy suorittamaan sille asetetun tehtävän. On tärkeitä ominaisuuksia, joita ilman ohjelma ei toimi ja on ominaisuuksia, jotka helpottavat ohjelman käyttöä ja tekevät siitä paremman.

2.3.1 Opinnäytetyöhön kuuluvat ominaisuudet

Tärkeisiin ominaisuuksiin kuuluu jonkinlainen pohja, jolle suunnitelma rakennetaan ja kyky lähettää se yritykselle, jotta se voi käydä asiakkaan tekemän suunnitelma läpi ja lähettää mahdollisen tarjouksen asiakkaalle takaisin. Tähän tietenkin tarvitaan myös tapa, jolla asiakas voi kertoa kuka hän on ja miten häneen saa yhteyttä, eli siis etunimi sukunimi ja sähköposti. Jonkinlaiset nimi- ja sähköpostikentät ovat erittäin tärkeitä ominaisuuksia. Tärkeäksi ominaisuudeksi voi myös laskea sen, että suunnitelmasta pystyy ottamaan palasia pois joko perumalla viimeisimpiä toimintoja, poistamalla paloja erillisen nappulan kanssa tai mahdollisuudella korvata paloja jälkeinpäin asettamalla uuden palan sen päälle. Parhaassa tapauksessa molemmat ominaisuudet ovat tietenkin käytettävissä.

2.3.2 Opinnäytetyön ulkopuolelle jäävät ominaisuudet

Näitä ominaisuuksia ovat esimerkiksi suunnitelman kolmiulotteinen mallinnus, automaattisen hinta-arvion laskeminen ja automaattisesti kääntyvät osat, jotta osia ei voi laittaa väärin päin. Nämä ominaisuudet olisivat suotavia sisällyttää ohjelmaan, jotta ne loisivat ohjelmalle paremman ulkoasun. Niitä ei kuitenkaan sisällytetä opinnäytetyön puitteisiin, mutta kerron niistä tässä hieman silti, koska se on ohjelman suunnittelun kannalta hyödyllistä.

Vaikein ominaisuuksista on kolmiulotteinen mallinnus asiakkaalle, josta hän hahmottaa paremmin rakennuksen. Tähän liittyen asiakas voisi myös vaihtaa seinien värejä tai kokeilla miltä erilaiset kattomateriaalit näyttäisivät. Jos ideaa haluaisi lähteä kehittämään vielä pidemmälle, voisi sisällyttää suunnitteluohjelmaan myös huonekalujen asettelun, jolloin asiakas voisi sisustaa suunnitelmansa samalla kun suunnittelee rakennustaan.

Automaattinen hinta-arvio laskettaisiin siten, että osilla on jokaisella omat hintansa ja ne laskettaisiin yhteen. Tämän lisäksi lattiat ja kattomateriaali nostaisivat hintaa. Mahdollista olisi myös, että palojen hinta laskee pikkuhiljaa tiettyyn pisteeseen asti sen mukaan mitä isomman rakennuksen asiakas on suunnitellut.

Automaattisesti kääntyvät osat toimisivat siten, että joka kulmalle ei tarvittaisi erilaisia osia niitä ei tarvitsisi manuaalisesti kääntää, vaan ne ikään kuin magnetisoituisivat oikein päin. Tämä vähentäisi huomattavasti tarvittavien osien määrää.

2.4 Työkalut

Työnteon apuvälineinä käytin Notepad++:a, Photoshop CS5:ttä, MSPaint-ohjelmaa, pikseligrafiikka ohjelmaa PyxelEdit ja Wacomin piirtopöytää. Lisäksi luonnostelin mahdollisia piirustuksia käyttöliittymästä perinteisesti kynällä ja paperilla.

2.4.1 Tekstieditori

Olin kahden vaiheilla siinä, että käytätkö tekstieditorina PsPad:ia vai Notepad++:a. PsPad-tekstieditoria oli opetettu käyttämään koulussa, mutta omaa kokemusta löytyi enemmän Notepad++:sta. Oli myös pidempi aika, kun olen viimeksi käyttänyt PsPadiä, sekä internetistä löytyvät tutorialit ja tieto oli suurimmaksi osaksi tehty Notepad++:lla. Tästä johtuen päätin valita Notepad++:n tekstieditorikseni.

Tekstieditoreilla ei ole suurempia eroavaisuuksia toiminnallisuudeltaan, mutta käyttöliittymät ovat jotakuinkin erilaiset. Erillisen tekstieditorin käyttämisen hyöty verrattuna tavallisen tekstitiedoston muokkaamiseen

on se, että tekstieditori värjää koodia sen mukaan mikä osa koodia se on, näyttää rivit, jolloin se ilmoittaa mahdollisista virheistä ja puuttuvista merkeistä.

2.4.2 Graafiset työkalut

Graafista työkalua ei tarvinnut valita, vaan hyödynsin tarvittaessa tilanteeseen parhaiten soveltuvaa graafista ohjelmaa. Käytin Painttia silloin kun tarvitsi nopeasti leikata kuva pienemmäksi tai rajata kuvakaappauksesta tietty alue näytettäväksi. Jotkin yksinkertaiset mallit on myös tehty Paintilla.

PyxelEdit on pikseligrafiikkaohjelma, jota olen käyttänyt paljon viime aikoina. Tällä ohjelmalla loin nopeasti väliaikaiset kuvat elementtien pohjapiirustuksista. Ohjelmalla pystyi helposti tekemään ruudukon ja tallentamaan jokaisen ruudun kuvat erillisinä tiedostoina kansioon, josta niitä oli helppo käyttää talosuunnitteluohjelmassa korvaavina kuvina.

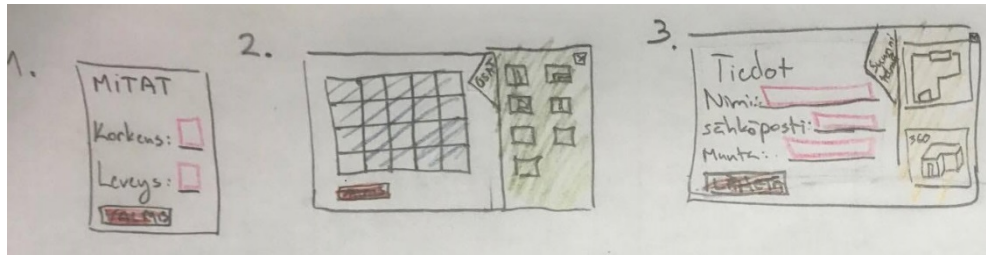
Photoshop CS5 versio, jota käytän, on vanha versio Photoshopista, jonka joskus ostin opiskelija-alennuksella. Se oli myös valmiiksi asennettuna tietokoneelleni, joten päätin käyttää sitä joidenkin luonnosten tekemiseen ja lopullisten pohjapiirustus palojen luomiseen. Photoshopia käyttäessäni minulla oli useimmiten käytössäni myös Wacom Intuos4 piirtopöytä.

2.5 Ohjelmien ulkoasun suunnittelu

Ennen kuin lähdin tekemään ohjelmaa, tein paperille muutaman suunnitelman siitä, miltä ohjelman tulisi suurin piirtein näyttää. Osa suunnitelmista on tehty sellaisiksi, että niiden kaikki ominaisuudet eivät sisälly oppinäytetyön puitteisiin. Suunnitelmia on tehty eri vaiheissa tutkimusta, joten ominaisuuksien määrä vaihtelee myös sen mukaan.

2.5.1 Ensimmäinen suunnitelma

Ensimmäinen suunnitelma oli, että ohjelma kysyisi ensin rakennuksen mitat ja loisi sopivan ruudukon sitä varten. Toisessa vaiheessa ruudulla olisi halutun kokoinen ruudukko näkyvissä ja oikeasta kulmasta saisi vedettyä esiin valikon, joka sisältää kaikki elementtiosat. Näillä elementtiosilla asiakas voisi luoda suunnitelmansa ja kun hän on valmis hän voisi siirtyä seuraavaan vaiheeseen. Kolmannessa ja viimeisessä vaiheessa asiakas laitetaan täyttämään omat tiedot, joihin sisältyy oma nimi, sähköposti ja vapaa kenttä omille kommenteille tai huomautuksille. Asiakas voi myös tarkastella suunnitelmaansa tässä vaiheessa. Mahdollista olisi myös tarkastella suunnitelmasta luotua kolmiulotteista mallia, jota asiakas voi katella eri kulmista (kuva 1).



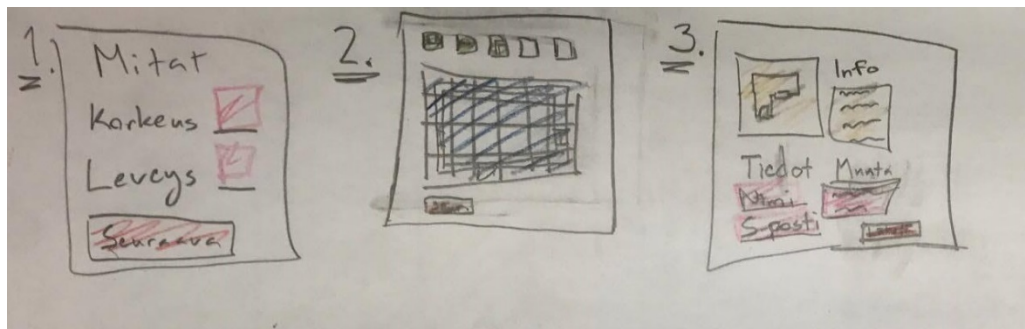
Kuva 1. Kuva ensimmäisen suunnitelman piirustuksesta.

Värikoodit kuvalle:

- Vaaleanpunainen: Asiakkaalle täytettävä kenttä.
- Vihreä: Liikkuva paneeli joka tulee ruudun päälle. Sisältää listan elementtiosista, joita käyttäjä voi siirtää ruudukolle. Paneeli menee takaisin piiloon, kun käyttäjä nappaa hiirellä elementtiosasta kiinni.
- Sininen: Suunnittelua varten tehty ruudukko.
- Punainen: Valmis nappula. Siirtää käyttäjän seuraavaan vaiheeseen. Viimeisessä vaiheessa nappulan nimeksi muuttuu läheta, joka painettaessa lähettää tarjouspyynnön, suunnitelman ja tiedot.
- Keltainen: Sisältää asiakkaan luoman suunnitelman ja kolmiulotteisen mallinnuksen asiakkaan suunnittelema rakennuksesta.

2.5.2 Toinen suunnitelma

Toinen suunnitelma oli huomattavan samanlainen ensimmäiseen suunnitelmaan verrattuna. Siinä kysyttiin ensin pituus ja leveys mittoja, joiden jälkeen käyttäjä siirtyy toiseen vaiheeseen, jossa on valmis ruudukko aikaisemmin syötetyillä mitoilla. Tässä vaiheessa erona on kuitenkin valmiiksi luodut seinät. Ruudukon uloimmilla ruuduilla on valmiiksi seinät ja käyttäjällä on mahdollisuus lisätä ovia ja ikkunoita ruudukon yläpuolelta haluamillensa kohdille. Kolmas vaihe on myös hyvin samanlainen kuin ensimmäisen suunnitelman vastaava vaihe, mutta kaikki tiedot näkyvät samaan aikaan ruudulla. Tässä versiossa ei ole kolmiulotteista mallinnusta rakennuksesta, vaan sen tilalla on muita tietoja rakennuksesta, kuten katon materiaali, rakennuksen mitat ja mahdollisesti jonkinlainen hinta-arvio (kuva 2).



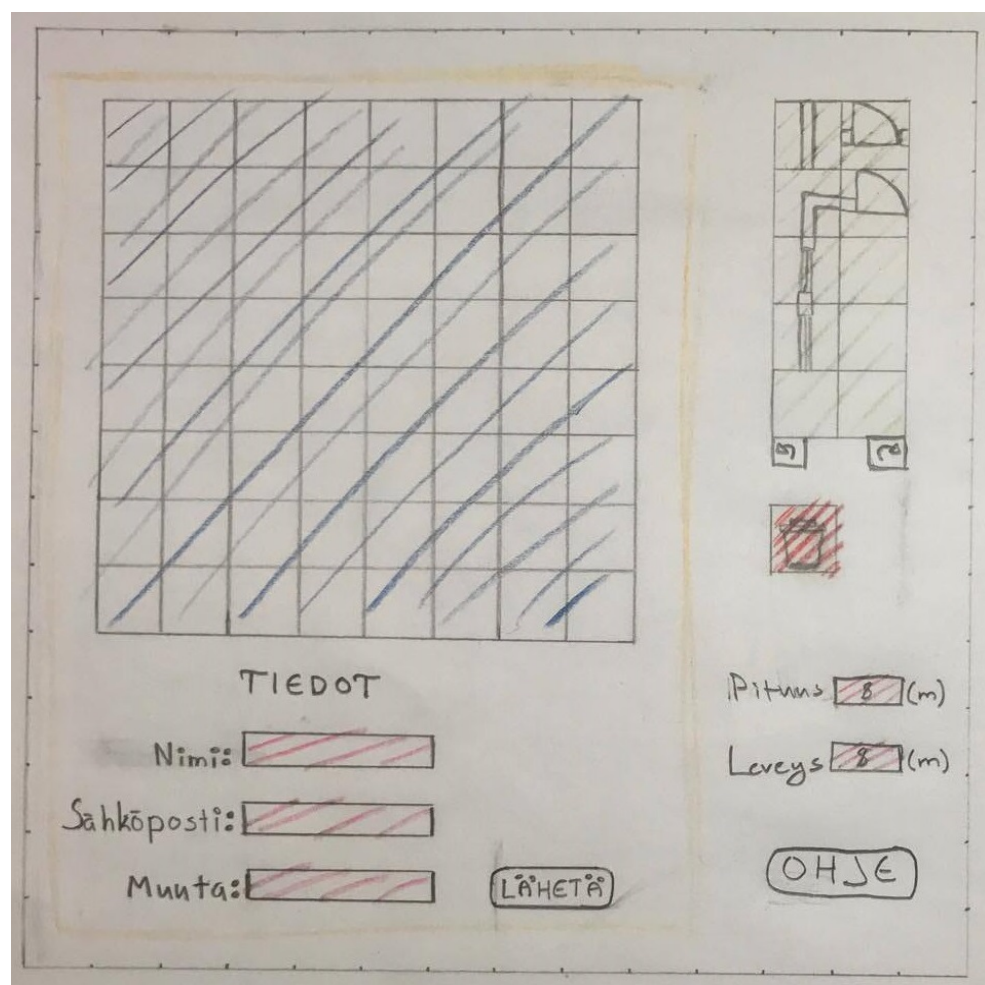
Kuva 2. Kuva toisen suunnitelman piirustuksesta.

Värikoodit kuvalle:

- Vaaleanpunainen: Asiakkaalle täytettävä kenttä.
- Punainen: Seuraava nappula. Siirtää käyttäjän seuraavaan vaiheeseen. Viimeisessä vaiheessa muuttuu lähetä nappulaksi, joka lähettää asiakkaan luomasta suunnitelmasta tarjouspyynnön.
- Vihreä: Ikkuna, ovi ja muut mahdolliset elementtiosat, joita käyttäjä siirtää ruudukon reunoille.
- Keltainen: Asiakkaan luoma suunnitelma ja tietoja suunnittelusta rakennuksesta, kuten mitat ja kattomateriaali.
- Sininen: Ruudukko, jonka uloimmille reunoille on valmiiksi luotu seinät.

2.5.3 Kolmas suunnitelma

Kolmas ja viimeinen suunnitelma oli lähimpänä lopullista tulosta. Se oli myös suunnitelma, jonka valitsin. Kolmannesta suunnitelmasta vaikutti todennäköisimmältä saada valmiiksi toimiva prototyyppi, jossa on suurin osa vaadittavista toiminnoista. Kaikki toiminnot ja tieto on myös sisällytetty samaan ikkunaan, ettei ole minkäänlaista vaiheistusta joka voisi sekoittaa asiakkaan ajatuksenjuoksun (kuva 3).



Kuva 3. Kuva kolmannen suunnitelman piirustuksesta.

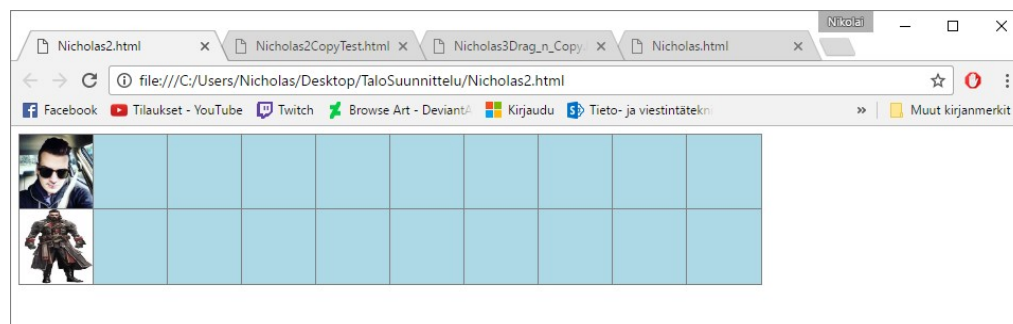
Sivuston käynnistyessä ruudulle pitäisi tulla ohjeilmoitus, joka sisältää perusohjeet ohjelman käyttämisestä ja tiedon lähettämisestä. Ohjeissa kerrotaan, että asiakkaan tulisi muistaa täyttää omat tietonsa ennen suunnitelman lähettämistä ja kehoitetaan valitsemaan rakennuksen mitat. Myös palojen siirtelystä, kääntämisestä ja poistamisesta neuvotaan. Ohje ikkunan saa esiin myös jälkeinpäin painamalla ohjenappulaa oikeassa alakulmassa.

Seuraava vaihe olisi pohjapiirroksen suunnittelu. Tässä vaiheessa käyttäjä siirtää elementtiosia ruudukolle ja luo oman suunnitelmansa. Palikoita voi käännellä valikon alla olevien painikkeiden avulla, mutta kääntäminen täytyy tehdä ennen kuin palikan siirtää ruudukolle. Näin säästyisi arvokasta tilaa, eikä tarvitse tehdä samasta palikasta montaa eri versiota. Valikon alapuolella on myös roskakorinappula, johon tarvittaessa voi siirtää väärälle paikalle menneen elementtiosan.

Viimeiseksi asiakas täyttää tiedot kohdan, syöttämällä tekstikenttiin oman nimensä, sähköpostinsa ja mahdollista muuta huomautettavaa. Lähetä-nappula ei suostu lähettämään, mikäli sähköpostia ei ole asetettu tarvittavaan kenttään.

3 TOTEUTUS

Ensimmäisenä loin perus pohjan HTML5-tiedostolle, joka sisälsi doctypen, sekä html-, head-, script-, meta-, ja body-tägit. Tämän jälkeen lähdin kehittämään web-ohjelmaa HTML5:lla tekemällä pari riviä ruutuja div-tägiä käyttäen. Tein näille diveille myös erikseen tyylin, vielä tällä hetkellä piittaamatta css-tyylitiedoston luomisesta. Lisäsin myös ensimmäisiin ruutuihin vasemmalla erilaiset kuvat, jotta voisin testata drag- ja drop-toimintoja seuraavaksi. Näiden jälkeen työ oli saatu hyvin alkuun ja tulos oli seuraavanlainen (kuva 4).



Kuva 4. Ensimmäinen testi ohjelmasta.

Kun olin saanut pohjan kuntoon, oli aika alkaa työstää toimintoja. Aloitin lisäämällä kuviin drag- ja drop-ominaisuudet hyödyntämällä seuraavanlaisia toimintoja script-tägin sisällä.

```
function drop(ev)
{
    ev.preventDefault();
    var data=ev.dataTransfer.getData("Text");

    var clone = document.getElementById(data).cloneNode(true);
    clone.id = clone.id + (new Date()).getMilliseconds();
    ev.target.appendChild(clone);

    if (ev.target == document.getElementById("trash"))
        var img = document.getElementById(data);
        img.parentNode.removeChild(img);
}
```

Kun olin saanut laitettua toiminnot script-tägien sisään, piti myös img-tägeihin lisätä drag(event)-toiminnot, jotta ne toimivat niin kuin niiden pitäisikin. Jokaisella div-tägillä tuli olla myös oma id ja niihinkin piti lisätä drop(event)- ja allowDrop(event)-toiminnot. Näin ollen kuvia pystyi siirtämään ruutujen välillä vaivatta.

3.1 Drag and copy

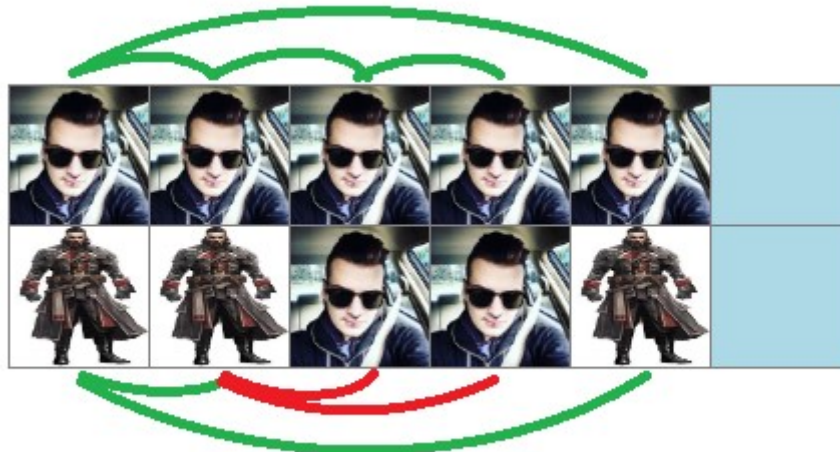
Seuraava vaihe oli muuttaa drop-toiminto "siirtämisen" sijasta siihen, että ohjelma kopioisi alkuperäisen kuvan. Tällöin olisi mahdollista luoda erillinen rivi talon elementeille, ikään kuin valikko, josta voi tuoda osia pohjapiirustusta varten ruudukolle.

Kopioimistoiminnon saavuttamiseksi, drop-toimintoa piti lähteä muokkaamaan siten, että se ei hae kuvaa pois alkuperäisestä sijainnistaan, vaan luo siitä kopion uuteen ruutuun. Tämä oli ensimmäinen kokeilu ruudun tiedon kopioimisesta, sen sijaan että sen siirtäisi. Alla olevan koodin toiminto siis korvasi ylemmän kuvan drop-toiminnon.

```
function drop(ev) {
    ev.preventDefault();
    var
    data=ev.dataTransfer.getData("text/html");
    var nodeCopy =
    document.getElementById(data).cloneNode
    (true);
    nodeCopy.id = "newId";
    ev.target.appendChild(nodeCopy);
}
```


}

Ensimmäinen tiedon kopioimiskokeilu toimi suhteellisen hyvin. Pystyin kopioimaan molemmat kuvat ja alkuperäinen lähde pysyi koskemattomana. Tässä tuli kuitenkin pieni epäkäytännöllisyys. Mikäli oli jo kopioinut kuvan yksi, ei voinut kopioida enää kuvaa kaksi kopioidusta lähteestä, vaan tähän ruutuun tuli kuva ensimmäisestä kuvasta (kuva 5).



Kuva 5. Viivat osoittavat mistä mihin kopioitu. Vihreä on toivottu tulos ja punainen ei toivottu tulos.

Tämä pieni epäkäytännöllisyys tulisi korjata heti, joten lähdin selvittämään mistä se johtui. Hetken pohdinnan ja kokeilun jälkeen sain selville, että poistamalla yhden rivin drop-toiminnosta, sain sen toimimaan halutulla tavalla. Alla oleva rivi on pois suljettu koodirivi.

```
/*nodeCopy.id = "newId";*/
```

En kuitenkaan poistanut riviä, vaan lisäsin rivin ympärille `"/* */` merkin, jotta sain suljettua rivin pois toimivasta koodista. Laitoin viereen myös selityksen, että toiminto tekee tehtävänsä paremmin ilman tätä riviä, mahdollista myöhempää selvittelyä varten.

Tämän jälkeen kuvat kopioituivat ruutuihin halutulla tavalla.

3.2 Ruudukko

Loin ruudukon div-tägejä käyttämällä ja kopioimalla niitä useaan kertaan peräjälkeen. Sijoitin div-tägit useamman div-tägin sisään ja asetin eri div-tägeille eri luokat, joille on määritetty tietyt tyyliasetukset css-tyylitiedostossa. Div-tägeihin lisättiin myös jo aikaisemmin käytetyt `drop(event)`- ja `allowDrop(event)`-toiminnot, jotta kuvien kopioiminen niihin onnistuisi.

```

<div class="grid">
  <div class="row">
    <div class="cell"ondrop="drop(event)"
ondragover="allowDrop(event)"></div>
    <div class="cell"ondrop="drop(event)"
ondragover="allowDrop(event)"></div>

```

Tämän jälkeen tein css-tyylitiedoston, jotta div-tägeille saa helposti tyyli-luokan johon viitata. Tämä helpottaa myös tulevaisuudessa uusien sa-manlaisten divien luomista ja tekee html-tiedoston sisällöstä helppolu-kuisempaa.

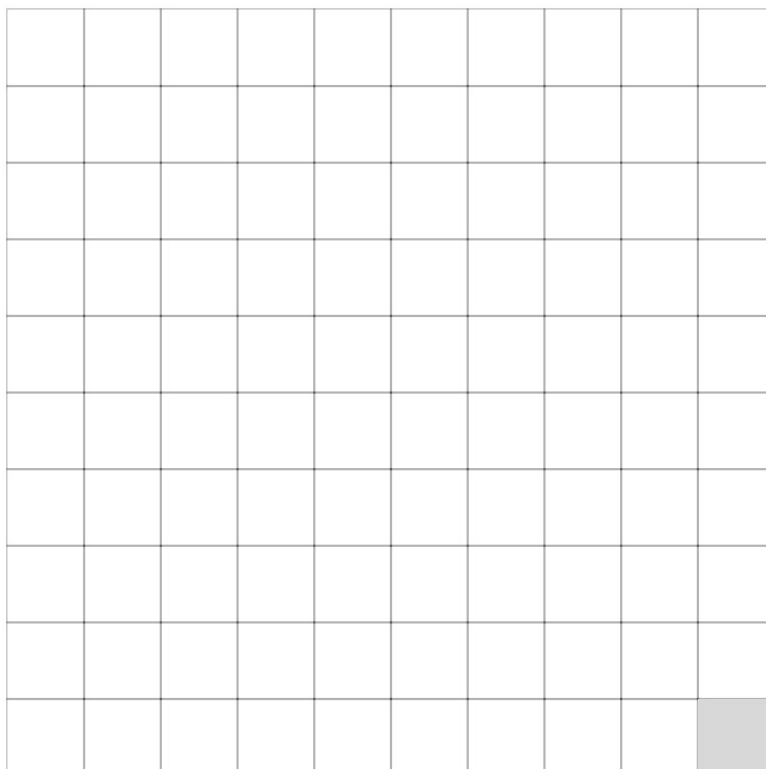
```

.grid > .row {
  font-size: 0;
  width: 100px;
  white-space: nowrap;
}

.grid > .row > .cell {
  position: relative;
  display: inline-block;
  width: 70px; height: 70px;
  outline: 1px solid rgba(0, 0, 0, 0.3);
}

```

Kun yllä olevan tyyliasetusten grid- ja row-osia käyttää div-tägien luokkia luodessa, saadaan siitä aikaiseksi seuraavan lainen ruudukko (kuva 6).



Kuva 6. Kuva div-tägeillä ja css-tyylitiedostolla luodusta ruudukosta.

Ruudukon oikean alakulman ruudun päällä on pidetty hiirtä ja tästä syystä ruutu on värjätty harmaaksi (kuva 6). Css-tyylitiedoston viimeinen rivi aiheuttaa ruudun harmaaksi muuttumisen, jotta käyttäjän on helpompi seurata hiiren liikettä, kun hän kopioi ruudulle palaa. Alla koodiesimerkki, joka muuttaa ruudun väriä.

```
.grid > .row > .cell: hover {
  background-color: #D8D8D8;
}
```

3.3 Prototyyppipalat

Seuraavaksi tein yhden ylimääräisen rivin erilliseksi kaiken yläpuolelle ja tämän rivin alapuolelle pienen marginaalin, jotta olisi helppo erottaa palat, joita on tarkoitus kopioida ruudukolle.

Loin pikaisesti PyxelEdit-ohjelmassa luonnokset ohjelman elementtiosista (kuva 7), joita voisi käyttää havainnollistavina kuvina lopullisten sijaan. Tässä vaiheessa tyydyin kuitenkin luomaan kuvat vain ovelle, ikkunalle, seinälle ja kulmapaloille.



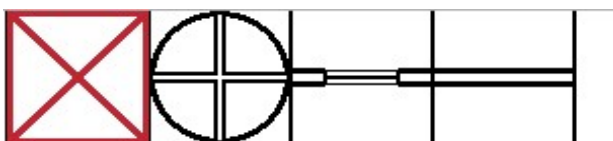
Kuva 7. Ensimmäiset versiot elementtiosien kuvista.

Seuraavaksi huomasin, että kopioitavat palat ja pohjapiirroksen ruudukko menevät päällekkäin, joten jouduin lisäämään wrapperin css-tyylitiedostoon ja laittamaan kaikki div-tägit wrapper divin sisälle, jolle oli asetettu seuraavanlaiset asetukset.

```
.wrapper {
  position: fixed;
  width: 100%; height: 100%;
  left: 0; top: 0;
  text-align: left;
}
.wrapper:before {
  content: "";
  display: inline-block; vertical-align: middle;
  width: 0; height: 100%;
}
.wrapper > .grid {
  display: inline-block;
  vertical-align: middle;
}
```

3.4 Poistaminen

Tässä vaiheessa on mahdollista luoda pohjapiirustuksia elementtirakennuksista. Ongelmaksi osoittautui kuitenkin, että mikäli teki virheen, palan poistaminen tai korvaaminen ei ollut mahdollista. Tässä tapauksessa ainoaksi mahdollisuudeksi jäi päivittää sivu ja aloittaa uudelleen tyhjästä. Tätä varten tein poistamiselle oman toiminnon, jotta käyttäjä pystyy poistamaan haluamansa osan siirtämällä sen poistamispainikkeen (kuva 8) päälle.



Kuva 8. Punainen rasti on poistamispainike.

Punaisen rastin kuva on luotu samalla tavalla kuin muutkin, paitsi että sille ei ole lisätty drag-toimintoa. Sen sijaan sille on lisätty drop-toiminto ja sen id:ksi on asetettu "trash". Seuraava koodi aiheuttaa sen, että mikäli kuvan pudottaa ruudulle jonka id on trash, se poistaa kuvan ruudusta eikä luo uutta kopiota kuvasta.

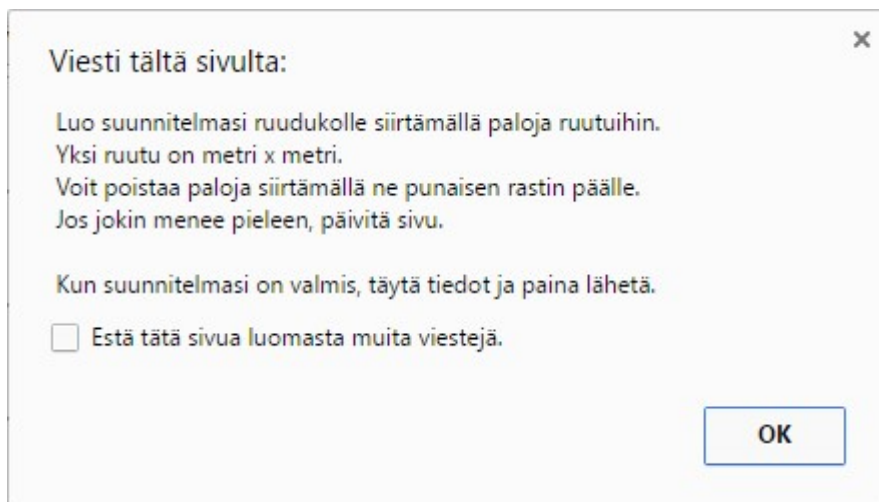
```
if (ev.target ==
    document.getElementById("trash")) {
    var img = document.getElementById(data);
    img.parentNode.removeChild(img);
}
```

3.5 Informaatio kentät

Informaatiokenttien luominen oli yksinkertainen prosessi. Loin kolme eri input-tägiä. Kaikkien näiden input-tägien tyypiksi asetin "text". Nimesin ne name-, email- ja info-nimillä, jotta voisin erottaa ne toisistaan. Asetin jokaiselle myös oman placeholderin, jossa lukee lisätietoa kyseisestä tekstikentästä (kuva 9).

Kuva 9. Tekstikentät.

Tämän jälkeen loin myös pikaisesti alle kaksi nappulaa button-tägiä käyttäen. Asetin ensimmäiseen nappulaan onclick-toiminnon, joka ilmoittaa ponnahdusikkunalla että ”Suunnitelmasi on lähetetty tarjouspyyntöä varten.”. Toinen nappula on ohjenappula, joka luo ponnahdusikkunan, jossa on ohjeita ohjelman käyttämistä varten (kuva 10).



Kuva 10. Ohje nappulan ponnahdusikkuna.

3.6 Tallentaminen

Suunnitelman tallentaminen osoittautui hiukan hankalaksi. Ensin selvitin, että miten sen tallentaminen onnistuisi helpoiten. Yritin ensin kaapata sivustosta kuvan html2canvas-nimisellä JavaScript-koodilla, mutta sen soveltaminen osoittautui hankalaksi, joten lähdin etsimään vaihtoehtoista tallentamistapaa.

Hetken mietiskelyn ja etsimisen jälkeen löysin toiminnon, jolla pystyy tallentamaan sivun html koodia html päätteiseen tekstitiedostoon. Rajasin ruudukon ja info osion erilliselle diville, jonka id:ksi asetin ”content”. Tätä id:tä hyödynnetään myöhemmin html koodin lataamisen rajausta varten siten, että se lataa vain suunnitelman ja info-osion, eikä lataa valmiiksi asetettu paloja joita on tarkoitus kopioida ruudukolle.

Lisäsin script-osioon toiminnon, joka mahdollistaa html-sisällön lataamisen, joka tämän jälkeen luo siitä html-tiedoston.

```
function download() {
    var a = document.body.appendChild(
        document.createElement("a")
    );
    a.download = "export.html";
    a.href = "data:text/html," + document.getElementById("content").innerHTML;
    a.click();
}
```

Tämän jälkeen lisäsin kyseisen toiminnon lähetä nappulan onclick-ilmoituksen jälkeiseksi toiminnoksi lisäämällä download-toiminnon onclick tägin loppuun.

```
<button type="button" onclick="alert('Suunnitelmasi on lähetetty tarjouspyyntöä varten.');" download() ">Lähetä</button>
```

Tämä lataustoiminto onnistuu lataamaan listan elementtiosista, jotka asiakas on laittanut ruudukolle. Toiminto ei kuitenkaan onnistu lataamaan tekstikenttiin asetettua tietoa, joten lataus-toimintoon piti lisätä muutama rivi koodia lisää. Lisätyt koodirivit asetettiin ylimmäisiksi.

```
function download() {
    var name = document.getElementById("name").value;
    var email = document.getElementById("email").value;
    var info = document.getElementById("info").value;
    document.getElementById("print").innerHTML =
name+"<br>" + email + "<br>" + info;
    var a = document.body.appendChild(
        document.createElement("a")
    );
    a.download = "export.html";
    a.href = "data:text/html," + document.getElementById("content").innerHTML;
    a.click();
}
```

Nämä koodirivit hakevat tekstikenttien sisällön ja tulostavat ne tekstinä uusien div-tägien sisään, jonka id:nä on print. Nämä div-tägit sisällytettiin content id:n omaavien div-tägien sisään, mutt ei grid id:n omaavien sisään, etteivät ne sotkeennu grid tyyliasetusten seurauksesta.

3.7 Virheiden korjaus ja hienosäätö

Testailin ohjelmaa säännöllisesti sitä tehdessä ja kirjasin ylös ongelmia ja muita virheitä, jotta voisin lopuksi korjata ja viilata mahdollisia epäkohtia pois. Jos ongelma oli liian iso, korjasin sen heti, mutta sellaisia ongelmia ei tullut vastaan kovin montaa. Jos ongelma oli ylitsepääsemätön, piti toiminto useimmiten korvata kokonaan jollain muulla toiminnolla.

3.7.1 Valikon kopioituminen

Tämä oli yksinkertainen ongelma löytää ja ratkaisu siihen oli myös yksinkertainen. Ongelmana oli, että jos siirsi elementtipalan valikossa sijaitsevan ruudun alareunaan, se loi ylimääräisen kuvan valikkoon. Ongelmana oli, että aikaisemmassa versiossa kaikissa ruuduissa oli ollut `drop(event)`- ja `allowDrop(event)`-toiminnot käytössä. Tästä johtuen lopulliseenkin versioon oli vahingossa jäänyt tällaiset toiminnot valikon ruuduille. Ongelma korjaantui sillä, kun poisti `drop(event)`- ja `allowDrop(event)`-toiminnot ruuduista.

3.7.2 Valikon asettelu.

Valikossa oli myös hieman hienosäädettävää. Alussa valikko oli vaakatasossa, mutta huomasin nopeasti, että se ei ollut hyvä idea. Vaihdoin valikon suunnan pystysuoraksi, sillä tämä säästi huomattavasti tilaa ja valikko oli helpommin saatavilla.

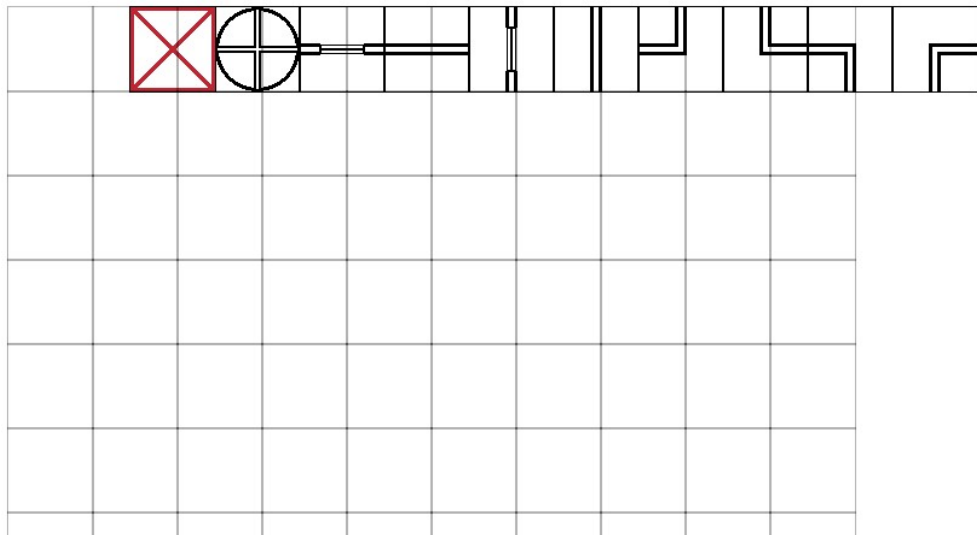
Valikon tyyli oli myös kirjoitettu html tiedostoon erillisen tyyliohjeiden sisälle. Tyylit oli tehty seuraavalla tavalla.

```
<style>
  #div0, #div1, #div2, #div3, #div4, #div5,
#div6, #div7, #div8 {
    float: right;
    width: 70px;
    height: 70px;
    margin: -1px;
    border: 1px solid black;
  }
</style>
```

Tämä oli ainut tyyliohje joka ei ollut css-tyylitiedostossa, joten päätin siirtää sen css-tyylitiedostoon. Tein myös hiukan muutoksia tyyliin samalla kun siirsin sen css-tyylitiedostoon.

```
.menu {
    float: right;
    width: 70px;
    height: 70px;
    margin: 20px;
}
```

Asetin samalla myös kaiken body-elementissä olevan sisällön div-tägien sisään, jolle annoin tyyliksi leveyden 850 pikseliä ja korkeudeksi 900 pikseliä. Tämä korjasi valikon asettumisen ruudukon päälle, sillä kun valikon tyylinä on "float: right;" se hakeutuu oikeaan reunaan ruudun koosta riippumatta (kuva 11).



Kuva 11. Vaakatasossa oleva valikko, joka hakeutuu oikeaan reunaan ja aiheuttaa ongelma olemalla ruudukon päällä.

Korjausten jälkeen lopputulos oli huomattavasti paremman näköinen ja helpommin käytettävä (kuva12).

Kuva 12. Lopullisen version pystyssä oleva valikko ei tule ruudukon päälle.

Huomasin että tyylitiedostossa olevalla wrapper-osuudella ei ollut mitään virkaa enää tässä vaiheessa, joten päätin poistaa kyseiset rivit tyylitiedoston koodista.

```
.wrapper {
  position: fixed;
  width: 100%; height: 100%;
  left: 0; top: 0;
  text-align: left;
}
.wrapper:before {
  content: "";
  display: inline-block; vertical-align: middle;
  width: 0; height: 100%;
}
.wrapper > .grid {
  display: inline-block;
  vertical-align: middle;
}
```

3.7.3 Elementtiosien kuvien linkitys

Kokeillessani ohjelmaa jossain muualla, huomasin siinä pienen vian, joka oli kohtalaisen helppo korjata. Olin linkittänyt elementtiosien kuvat seuraavalla tavalla.

```

```

Edellä mainitussa tavassa oli se ongelma, että jos kansion siirsi jollekin muulle koneelle, ei linkki pitänyt enää paikkansa. Tässä tapauksessa kuvaa ei siis löytynyt ja valikon ruudut jäivät tyhjiksi. Korjasin tilanteen sillä, että poistin src-tägin hapsujen sisältä tiedoston reitin ja jätin ainoastaan tiedoston nimen etsittäväksi. Tämän jälkeen kuvat löytyivät oikealla tavalla, vaikka web-sivua käytti eri tietokoneella ja koodikin näytti paljon siistimmältä, kuten alla olevassa esimerkissä näkyy.

```

```

4 YHTEENVETO

4.1 Lopputulos

Lopputuloksena sain aikaiseksi verkkosivun, jossa pystyy suunnittelemaan pohjapiirustuksen varastorakennukselle. Suunnittelun jälkeen, käyttäjä pystyy lataamaan listan suunnitelmaan tarvittavista osista html-tiedoston muodossa. Mikäli html-tiedoston sisällön sijoittaa web-sivun koodin "grid"-osion tilalle, se luo oikean näköisen suunnitelman asiakkaan tekemästä suunnitelmasta. Tämän automaatio olisi suotavaa, mutta se jää opinnäytetyön ulkopuolelle.

Lopputulos olisi ollut ainakin hieman kehittyneempi ja hiotumpi, mikäli olisin käyttänyt avoimen lähdekoodin kirjastoja. Näihin kuuluu esimerkiksi jQuery ja Bootstrap. Jos olisin tutustunut näihin heti alussa, olisi lopputulos todennäköisesti ollut toisenlainen. Toisaalta nytkään opittu asia ei mennyt hukkaan.

4.1.1 JQuery

Tutustuin jQueryyn tarkemmin vasta sen jälkeen, kun ei ollut enää järkevää lähteä käyttämään sitä web-ohjelmaa tehdessä. Minulla ei myöskään ollut lainkaan kokemusta jQueryn käytöstä, joten suljin sen vaihtoehtona pois tiedon ja osaamisen puutteen vuoksi.

Jälkeenpäin ajateltuna jQueryn käyttäminen olisi ollut viisasta, sillä iso osa ominaisuuksista olisi voitu kopioida jQuery-kirjastosta ja ominaisuuksista olisi täten tullut hiotumpia. JQueryä käyttämällä ohjelman kehitys olisi nopeutunut ja täten olisin saanut lisättyä siihen ominaisuuksia, joita jäi puuttumaan.

4.1.2 Bootstrap

Bootstrap on asia, jota tutkin myös tarkemmin vasta myöhemmin. Työn loppuvaiheilla opettaja mainitsi minulle tästä, jolloin ajattelin kuitenkin tutustua siihen. Mikäli olisin aikaisemmin tutustunut aiheeseen, olisi ohjelmasta todennäköisesti tullut käyttäjäystävällisempi ja miellyttävämmän näköinen.

4.2 Arviointi

Työn tekeminen sujui suhteellisen mutkattomasti, vaikka aika-arviot venyivät. Varsinkin suunnitteluvaiheessa, kun tutkin mahdollisia työkaluja ja toteutustapoja. Huomasin yhtäkkiä, että olen käyttänyt suuren määrän aikaa pelkästään suunnitteluun, vaikka itse toteuttamista ei ollut vielä

edes aloitettu. Pääsin kuitenkin tavoitteeseen suhteellisen hyvin, muutama ominaisuutta lukuun ottamatta.

Isoin virhe ehdottomasti opinnäytetyön suorittamisessa oli avointen lähdekoodikirjastojen ja koodi-kehysten käyttämättä jättäminen. Niiden käyttämättä jättäminen johtui siitä, että niistä ei ollut lainkaan aikaisempaa kokemusta tai opetusta. Tietämykseni niistä oli myös hyvin rajallista ennen työn aloittamista.

Koodikirjastojen ja -kehysten käyttämättä jättämisestä huolimatta lopputuloksesta tuli toimiva prototyyppi, joka sisälsi tarvittavat ominaisuudet. Opin projektia tehdessä paljon uutta web-tekniikoista, varsinkin JavaScriptistä ja sen erilaisista käyttötavoista. Tämä oli myös ensimmäinen käyttöliittymäsuunnittelu projekti, jonka olen vienyt alusta loppuun asti yksin. Sain paljon kokemusta ja tietotaitoa web-käyttöliittymän suunnittelusta, sekä monista eri työkaluista joita sitä tehdessä kannattaa hyödyntää.

LÄHTEET

Bootstrap (n.d.)

Haettu 19.4.2017 osoitteesta

<http://getbootstrap.com/>

Codepedia (n.d.)

Haettu 14.4.2017 osoitteesta

<http://codepedia.info/convert-html-to-image-in-jquery-div-or-table-to-jpg-png/>

Dive Into HTML5, Mark Pilgrim (2009)

Haettu 30.4.2017 osoitteesta

<http://diveintohtml5.info/>

Eloquent Javascript, Marjin Haverbeke (2014)

Haettu 8.4.2017 osoitteesta

<http://eloquentjavascript.net/>

HTML Canvas Deep Dive, Josh Marinacci (2012)

Haettu 29.4.2017 osoitteesta

<https://joshondesign.com/p/books/canvasdeepdive/toc.html>

HTML5: The Missing Manual, Matthew MacDonald (Joulukuu, 2013)

Haettu 9.4.2017 osoitteesta

https://books.google.fi/books?id=UwhaAgAAQBAJ&pg=PA6&dq=html5&hl=fi&sa=X&redir_esc=y#v=onepage&q=html5&f=false

jQuery (n.d.)

Haettu 20.4.2017 osoitteesta

<https://jquery.com/>

jQuery Fundamentals, Rebecca Murphey (2012)

Haettu 25.4.2017 osoitteesta

<http://jqfundamentals.com/>

Speaking JavaScript, Axel Rauschmayer (Helmikuu, 2014)

Haettu 21.4.2017 osoitteesta

<http://speakingjs.com/es5/>

Stackoverflow (n.d.)

Haettu 14.4.2017 osoitteesta

<http://stackoverflow.com/questions/29702758/html-button-to-save-div-content-using-javascript>

Stackoverflow (n.d.)

Haettu 6.2.2017 osoitteesta

<http://stackoverflow.com/questions/34144191/how-to-create-a-isometric-drag-drop-grid-with-javascript>

Stackoverflow (n.d.)

Haettu 6.2.2017 osoitteesta

<http://stackoverflow.com/questions/13007582/html5-drag-and-copy>

Stackoverflow (n.d.)

Haettu 1.2.2017 osoitteesta

<http://stackoverflow.com/questions/8140574/drag-picture-into-html-table>

Stackoverflow (n.d.)

Haettu 4.2.2017 osoitteesta

http://stackoverflow.com/questions/13007582/html5-drag-and-copy#comment41979767_13008017

Stackoverflow (n.d.)

Haettu 6.2.2017 osoitteesta

<http://stackoverflow.com/questions/27089260/drag-and-drop-replace-in-html-5>

Stackoverflow (n.d.)

Haettu 6.2.2017 osoitteesta

<http://stackoverflow.com/questions/22571483/how-to-drag-image-to-trash-can-and-remove-that-single-image>

W3schools HTML5 (n.d.)

Haettu 8.4.2017 osoitteesta

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

W3schools JavaScript (n.d.)

Haettu 18.2.2017 osoitteesta

https://www.w3schools.com/js/js_where.asp

W3schools jQuery (n.d.)

Haettu 30.4.2017 osoitteesta

<https://www.w3schools.com/jquery/>

What is Bootstrap and How Do I Use It?, Tania Rascia (Marraskuu, 2015)

Haettu 1.5.2017 osoitteesta

<https://www.taniarascia.com/what-is-bootstrap-and-how-do-i-use-it/>

Wikipedia Bootstrap (n.d.)

Haettu 19.4.2017 osoitteesta

[https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

Wikipedia CSS (n.d.)

Haettu 8.4.2017 osoitteesta

https://fi.wikipedia.org/wiki/Cascading_Style_Sheets

Wikipedia Front-end web development (n.d.)

Haettu 27.2.2017 osoitteesta

https://en.wikipedia.org/wiki/Front-end_web_development

Wikipedia HTML (n.d.)

Haettu 8.4.2017 osoitteesta

<https://fi.wikipedia.org/wiki/HTML>

Wikipedia HTML5 (n.d.)

Haettu 8.4.2017 osoitteesta

<https://fi.wikipedia.org/wiki/HTML5>

Wikipedia JavaScript (n.d.)

Haettu 6.4.2017 osoitteesta

<https://fi.wikipedia.org/wiki/JavaScript>