Jyri Mäntylä

FYSIOTERAPEUTTINEN KEHONKUNTOUTUSSOVELLUS

Tietojenkäsittelyn koulutusohjelma
2017

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

FYSIOTERAPEUTTINEN KEHONKUNTOUTUSSOVELLUS

Mäntylä, Jyri
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Joulukuu 2017
Ohjaaja: Hentunen, Ilmari
Sivumäärä:28
Liitteitä:0

Asiasanat: Web-sovellus, full-stack, palvelin, suunnittelu, ohjelmointi

---

Opinnäytetyössä tarkastellaan sovelluksen kehitysprojektia, jossa määriteltiin, suunniteltiin ja toteutettiin fysioterapeuttinen sovellus, joka huoltaa toimistotyötä tekevien ihmisten fysiologisia ongelmia. Oman työni pääpaino oli sovelluksen määrittelyssä, suunnittelussa ja toteuttamisessa tietojenkäsittelyn näkökulmasta. Työ tehtiin nykyisen fysioterapeutti Niklas Seppälän kanssa. Niklas hoiti sovellusidean pohtimisen, suunnittelemisen ja toteuttamisen fysioterapeutin näkökulmasta.

Sovellus muistuttaa käyttäjää huoltamaan kehonsa osia ja ohjeistaa häntä videoiden avulla. Sovelluksessa valitaan kuntoutettavat kehon osat ja muistutuksen ajankohta. Muistuttaminen tapahtuu sähköpostin avulla, joka saa puhelimessa aikaan äänen haluttuna aikana ja tarjoaa suoran linkin kuntouttavaan videoon. Sovellus on saatavissa osoitteessa: http://jobergo.jyrin.com/

PHYSIOTHERAPEUTIC HEALING SOFTWARE FOR OFFICE EMPLOYEES

Mäntylä, Jyri
Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences
Degree Programme in Business Information Technology
December 2017
Supervisor: Hentunen, Ilmari
Number of pages:28
Appendices:0

Keywords: Web application, full-stack, server, design, programming

---

This Bachelor's Thesis examines the software development project, consisting of software requirements, design and implementation of a physiotherapeutic application for the physiological problems of office workers. The main focus of my work was to determine, design and implement the application from the point of view of information technology. The work was done with the current physiotherapist Niklas Seppälä. Niklas took care of thinking, designing and implementing an application idea from a physiotherapist point of view.

The app reminds the user to care for the parts of his body and to guide him with physiotherapist videos. In the application user selects the rehabilitation of body parts and the time of reminder. Reminders are sent via e-mail, which makes alarm sound at the desired time and provides a direct link to a rehabilitating video at user’s phone. The application is available at: http://jobergo.jyrin.com/

SISÄLLYS

1 JOHDANTO

1.1 Työn tausta ja tarkoitus

Tässä opinnäytetyössä tarkastellaan web-pohjaista kuntoutustyökalua, joka on luotu fysioterapeuttisen hoidon apuvälineeksi. Kuntoutussovellus auttaa kuntoutettavaa pitämään yllä kuntoutusrutiinia ja tarjoaa harjoitteita paikasta ja vuorokaudenajasta riippumatta. Oman työni pääpaino oli sovelluksen suunnittelussa ja toteuttamisessa tietojenkäsittelyn näkökulmasta. Työ tehtiin nykyisen fysioterapeutti Niklas Seppälän kanssa. Niklas hoiti sovellusidean suunnittelemisen ja toteutuksen fysioterapeutin näkökulmasta, eli ideoi ja toteutti tietosisällön, mukaan lukien videot. Hän myös teki sovelluksen valmistuttua käyttötestit pienryhmälle.

Sovellus muistuttaa käyttäjää huoltamaan kehonsa osia ja ohjeistaa häntä videoiden avulla. Sovelluksessa valitaan kuntoutettavat kehon osat ja muistutuksen aika. Muistuttaminen tapahtuu sähköpostin avulla, joka saa Android puhelimessa aikaan äänimerkin haluttuna aikana ja tarjoaa suoran linkin kuntouttavaan videoon.

1.2 Tutkimusongelma

Tutkimusongelma on määritelty kysymysmuodossa seuraavasti: miten rakentaa helposti lähestyttävä, heti kokeiltavissa oleva ja laitteesta riippumaton kehonhuolto-sovellus? Kysymys muotoutui fysioterapeutti Niklas Seppälän ideasta ja molempien halusta rakentaa kehonhuoltosovellus, joka olisi joillain erityisillä tavoilla parempi kuin muut vastaavat sovellukset. Muita vastaavia sovelluksia vertailtaessa huomattiin, että niillä oli heikkouksia kuten laitteistoriippuvuus ja vaikea lähestyttävyys. Nämä heikkoudet tulivat ilmi kahdessa suositussa ohjelmassa kuten Ergoprossa ja Office ergonomics softwaressa. Ergopro on sidottu laitteeseen ja käyttöjärjestelmään, eikä sen toimintaa opi tuntemaan muuta kuin ottamalla yhteyttä asiakaspalveluun ja

kysymällä testiversioita (Ergopro 2017). Dokumentaatio sovelluksen internet-sivuilla oli hyvin vähäistä. Samat vaikeudet jatkuivat Office ergonomics softwaressa, eikä siitäkään löydetty järkevässä ajassa dokumentointia (Office ergonomics software 2017). Toisin sanoen molempien ohjelmien kokeileminen oli tehty käyttäjälle hankalaksi. Näin päädyttiin tutkimusasetelmaan, jossa tarkoituksena on rakentaa helposti lähestyttävä, heti kokeiltavissa oleva ja laitteesta riippumaton kehonhuolto-sovellus.

1.3 Opinnäytetyön rakenne

Toisessa luvussa tarkastellaan web-järjestelmän määrittelyä ja suunnittelua. Tämän jälkeen käydään läpi sovelluksen toiminnan kannalta oleellisimmat ohjelmointikielet ja kirjastot. Neljännessä luvussa tutustutaan työkaluihin, joiden avulla sovellus on ohjelmoitu. Lisäksi käydään läpi palvelinympäristön pystyttämistä, jonka päälle sovellus on rakennettu. Palvelinympäristö ja sovellus yhdessä muodostavat web-järjestelmän. Viidennessä luvussa kerrotaan sovelluksen toteutuksesta. Työn lopuksi esitetään yhteenveto.

2 WEB-JÄRJESTELMÄN MÄÄRITTELY JA SUUNNITTELU

2.1 Vaatimusmäärittely

Vaatimusmäärittely sisältää vaatimusten esilletuomisen, analysoinnin, spesifioinnin ja validoinnin sekä vaatimusten hallinnoinnin ohjelmiston elinkaaren ajan. Yleisesti tunnustetaan, että ohjelmistoprojekti on hyvin haavoittuvainen jos vaatimukset on heikosti määritelty. Määrittely ilmaisee ohjelmiston tarpeet ja rajoitteet ohjelmiston kehittämiseksi. Vaatimusmäärittely on koko ohjelmiston elinkaaren ajan jatkuva prosessi, eikä päättyvä osa ohjelmistokehityksen alussa. Vaatimuksia on erilaisia ja ne ovat tarpeita tai rajoitteita ohjelmiston kehitykselle, joita voidaan luokitella

toiminnallisiin, ei-toiminnallisiin ja mitattaviin. (Bourque & Fairley 2014.)

Vaatimusmäärittelyssä hahmotetaan siihen osallistuvat henkilöt kuten fysioterapeutti ja ohjelmisto-insinööri. Määrittelyssä tulisi ottaa huomioon sovelluskehittäjien taitotaso suhteessa vaatimuksiin, joka vaikuttaa projektin kustannuksiin ja aikatauluun. Kaikkia vaatimuksia ei voida täyttää ja on osallistujien tehtävänä neuvotella ja saavuttaa yhteisymmärrys kun vastakkaisia näkemyksiä nousee esiin. Fundamentaali periaate hyvässä vaatimusten esiintuonnissa on tehokas kommunikointi osallisten välillä. Tämän pitäisi jatkua koko ohjelmiston kehityskaaren ajan. Vaatimuksista puhutaan osallisten välillä yhteisesti sopivilla abstraktion tasoilla. Tärkeää on saada tietoon projektin pituus, kuvaus ohjelmistosta, sen tarkoituksesta ja toimitettavien vaatimusten prioriteetista. Liiketoiminnan kannalta tärkeimmät vaatimukset tulisi täyttää ensisijaisesti. (Bourque & Fairley 2014.)

Vaatimuksia saadaan tuotua ilmi haastattelujen, skenaarioiden, prototyypittämisen, tapaamisten, käyttäjätarinoiden, UML-notaatioiden ja kilpailijoiden tuotteiden analysoinnin avulla. Tärkeää olisi, että käytetyt tekniikat kuvaisivat vaatimuksia niin hyvin kuin mahdollista. Vaatimuksista tuotetaan ohjelmiston vaatimusmäärittelydokumentti, joka on asiakkaan ja tuottajan välillä lähtökohtana siitä mitä ohjelman tulee tehdä ja mitä siltä ei odoteta. Tällöin varmistetaan, että vaatimukset ovat ymmärrettäviä ja selkeitä ennen kuin niihin aletaan panostamaan resursseja. Tämä takaa vakaan pohjan suunnittelulle. Lopuksi tehdään hyväksymistestit, jolloin valmiin ohjelmiston tulisi vastata vaatimusmäärittely-dokumenttia. (Bourque & Fairley 2014.)

2.2 Jobergo-kehonkuntoutussovelluksen määrittelyä

Ensimmäiseksi fysioterapeuttisten tarpeiden osalta haluttiin sovellus jossa on kehonhuoltoa opettava osa ja kehonhuollosta muistuttava osa. Kehonhuoltovideoiden ja lisäinformaation avulla käyttäjää ohjeistava osa nimettiin taukojumppariksi.

Käyttäjää muistuttava osa nimettiin ryhtimuistuttajaksi, sen tulisi muistuttaa käyttäjää halutuin väli-ajoin. Fysioterapeuttiset perustelut taukojumpparin ja ryhtimuistuttajan tarpeelle kehonhuollossa löytyvät Niklaksen opinnäytetyöstä Jobergo hyvinvointisovellus: Motorisen kontrollin harjoitteiden käytettävyystestaus (Seppälä, 2016).

Toiseksi määrittelyssä tarkasteltiin Niklaksen kanssa muita vaihtoehtoisia sovelluksia ja pohdittiin mitä voidaan tehdä paremmin. Tutkimalla vastaavanlaisia tuotteita huomattiin, että niiden kokeileminen vaatii yhteydenottoa yritykseen ja sovelluksesta itsestään ei löytynyt kattavaa helposti löydettävää dokumentointia eli sovellukset olivat vaikeasti lähestyttävissä. Ainakin toinen sovellus oli riippuvainen Windows-käyttöjärjestelmästä. Näin löydettiin parannettavat ydinalueet eli sovelluksen tulisi olla helposti lähestyttävä, heti kokeiltavissa ja laitteisto-riippumaton.

Tavoitteina ”heti kokeiltavissa oleva” ja ”mahdollisimman laitteisto-riippumaton” mukailevat web-kehitystä. Nykyään kaikki tietokoneet ja puhelimet alkavat olla yhteydessä internetiin, joka mahdollistaa tavoitteiden mukaisen sovelluksen kehittämisen. Tavoite saavutetaan kun mikä tahansa internetin ja selaimen omaava laite kykenee käyttämään sovellusta ja muistuttajaa. Ryhtimuistuttaja toimii selaimen sähköpostin välityksellä ja taukojumppari toimii selaimessa. ”Heti kokeiltavuus” onnistuu kun ohjelman sivulle tullessa pystyy aloittamaan taukojumpparin käytön ja kirjautumaan ryhtimuistuttajaan. Näin määritellään kaikille avoin, heti kokeiltavissa oleva ja laitteistoriippumaton kehonhuolto-sovellus.

2.3 Suunnittelu

Ohjelmistosuunnittelu on prosessi, jossa määritellään systeemin arkkitehtuuri, komponentit, rajapinnat ja muut piirteet. Siinä vaatimusmäärittelystä rakennetaan kuvaus ohjelmiston sisäisestä rakenteesta sillä tasolla, että se pystytään suunnitteludokumenttien avulla toteuttamaan ja käyttämään apuna testauksessa. Suunnittelu on vaatimusmäärittelyn ja toteutuksen välinen vaihe, jossa arkkitehtuuri

on korkeammalla tasolla kuvaus komponenttien organisoinnista ja matalammalla tasolla kuvataan komponentin käyttäytyminen. Kuvauksien luomiseen voidaan käyttää graafisia ja tekstipohjaisia menetelmiä kuten UML-kaavioita ja pseudokoodia. Arkkitehtuurista rakennetta kuvataan usein rakennekaavioilla, jotka antavat kuvan ohjelmiston staattisesta rakenteesta. Matalammalla tasolla taas käytetään useammin käyttäytymiskaavioita tai pseudokoodia, joilla voidaan tarkemmin hahmottaa erilaisten yksittäisten komponenttien toimintaa ja vuorovaikutusta. Kuvauksien avulla ohjelmoijan tulee pystyä hahmottamaan ja toteuttamaan ohjelma. Kirjaamalla perustelut päätöksien takana ohjelmiston ylläpito helpottuu. (Bourque & Fairley 2014.)

Ohjelmistosuunnittelussa on useita pääperiaatteita, joita kaikki suunnittelustrategiat noudattavat ja niiden avulla ohjelman vaatimukset saadaan jalostettua toteutettavaan muotoon. Pääperiaatteita ovat vaatimusten abstraktointi, pilkkominen komponentteihin joille määritellään rajapinnat, joiden kautta määritellään komponenttien riippuvuudet. Komponenttien sisäiset yksityiskohdat määritellään ja ne pakataan niin, että ulkopuoliset entiteetit eivät pääse rajapintojen kautta käsiksi muihin sisäisiin yksityiskohtiin kuin mitä on tarkoitettu. Komponenttien tulisi sisältää vain oleellinen, eikä muuta. Edellämainittujen perjaatteiden avulla tulisi muodostaa helposti ymmärrettävä johdonmukainen kuvaus ohjelmasta. (Bourque & Fairley 2014.) Näiden pääperiaatteiden noudattaminen voi toteutua jotain valmista suunnittelustrategiaa hyödyntäen, kuten vaikkapa Jobergon tapauksessa jossa mukailtiin funktiopohjaista suunnittelua. Siinä ohjelma jaetaan päätoimintoihin ja niitä jalostetaan korkealta tasolta alaspäin tarkentaen. Jobergoa suunnitellessa käytettiin paljon pseudokoodia jossa korkeammalla tasolla kuvattiin yleisesti komponentteja kuten ryhtimuistuttajan ja taukojumpparin toimintaa ohjelmassa. Kerros kerrokselta niiden toimintaa tarkennettiin matalammalle tasolle käskyiksi. Myös datan sisääntuloa esimerkiksi lomakkeen kautta ja sen liikettä Crontabissa olevaksi ajastukseksi kuvattiin pseudokoodin avulla.

Ohjelmistosuunnittelun ja toteutuksen aikana on hyvä kiinnittää huomiota laatuseikkoihin kuten järjestelmän suorituskykyyn, tietoturvallisuuteen, vikojen ja

vikasietoisuuden hallintaan, käytettävyyteen ja datan hallintaan. Datan hallintaan liittyy esimerkiksi oikeanlaisen tietokannan suunnittelu, jossa selvitetään minkälainen tietokanta sopii sovelluksen rakenteeseen. Tietokantatyyppejä on useampia erilaisia, kuten relaatiotietokanta, lohkoketju ja tiedostotietokanta. Jokaisella tietokantatyypillä on omia hyviä ja huonoja puolia. Useimmin valitaan tiedostotietokannan tai relaatiotietokannan väliltä. Tiedostotietokanta on yksinkertainen tekstitiedostoon kirjoitettava taulu, jossa rivi sisältää yhden tiedon ja rivit erotellaan erotinmerkillä kuten pilkulla. Tiedostotietokantaan ei tarvitse tehdä ylimääräisiä valmisteluja verrattuna relaatiotietokantaan, kuten tietokanta-ohjelmiston asennusta ja tietokannan suunnittelua. Riittää kun ohjelmoija luo tiedoston, tallentaa siihen datan ja lukee sen PHP-scriptin metodilla (Valade 2011). Koska Jobergo-sovelluksen tarvitsi kirjata vain muistutettava aika ja sähköposti-osoite niin siihen sopi hyvin tiedostotietokanta, eikä tiedon relaatioille ollut tarvetta. Käytännössä tiedostotietokantana käytettiin Linux-käyttöjärjestelmässä olevaa Crontabia, jonne talletettiin muistutusajat. Toinen tietokanta sijaitsi sovelluksen juuri-kansion alikansiossa, jonne talletettiin sähköpostiosoitteet.

2.4 käyttöliittymä

Käyttöliittymän tulisi varmistaa että vuorovaikutus ihmisen ja koneen välillä on tehokasta, sekä vastata käyttäjien taitoja, kokemuksia ja odotuksia. Yleisesti käyttäjää tulisi pyrkiä huomioimaan tekemällä ohjelmistosta helposti opittava, säännönmukainen ja siinä tulisi käyttää termejä ja konsepteja, jotka ovat tuttuja käyttäjille sekä tarjota käyttö-opastusta ja mekanismeja ongelmien ratkaisemiseen sekä ottaa huomioon erityyppiset käyttäjät(Bourque & Fairley 2014). Syksyllä 2017 käymässäni keskustelussa Michael Fynesin(Futuremark) kanssa hän kertoi, että käyttöliittymäsuunnittelussa tulisi pyrkiä tekemään kaikesta mahdollisimman yksinkertaista ja se miten tähän päästään tulee olemaan monimutkaista.

Käyttöliittymää on hyvä suunnitella prototyyppien avulla. Tällöin käyttäjäkokemukset auttavat käyttöliittymän ominaisuuksien kehittämisessä. Esimerkiksi Jobergon

suunnittelussa prototyypitettiin paljon ja kokeiltiin ohjelman toimivuutta, josta saadun palautteen avulla tarkennettiin ja muokattiin sovellusta. Käytännössä muokattiin sitä miten tieto esitetään tai sitä miten käyttäjä vuorovaikuttaa yksityiskohtaisemmin ohjelman kanssa.

Tieto voidaan esittää tekstipohjaisesti tai graafisesti. Värien käytöllä voidaan vahvistaa tiedon esittämistä esimerkiksi kertomalla ohjelmiston statuksen muutoksesta tai johdonmukaisesti ohjaamalla käyttäjää vaikkapa yhdistämällä tietynlaiset toiminnot tiettyyn väriin. Toistamalla yleisiä käyttöliittymän ulkoasumalleja helpotetaan käyttäjää omaksumaan ohjelma kun tutut tiedot ja toiminnot löytyvät tutuilta paikoilta. Näiden toimintojen vasteaikojen ja palautteen suunnittelu pitää käyttäjän kartalla siitä missä mennään vaikkapa kertomalla käyttäjälle että toiminto on nyt suoritettu tai näyttämällä toiminnon etenemistä seuraava tietoa kuten videosoittimen aikajanaa. Käyttäjän ja ohjelman vuorovaikutus taas tapahtuu useimmin lomakkeiden käytöllä, valikoilla, suoralla objektin manipuloinnilla esimerkiksi hiirtä käyttäen, käskykielellä jossa käyttäjä kertoo käskyn ja antaa parametrit tai luonnollisella kielellä, joka käännetään koneelle ohjeiksi. (Bourque & Fairley 2014.)

3 OLEELLISIMMAT OHJELMOINTIKIELET JA KIRJASTOT

3.1 PHP

PHP on yksi käytetyimmistä ohjelmointikielistä ja sitä käytetään pääasiallisesti web-ohjelmoinnissa toiminnallisuuden luomiseen. Tärkein käyttöympäristö on palvelimella tapahtuva tietojenkäsittely ja tietokannan kanssa vuorovaikuttaminen. PHP:ta pystyy käyttämään toiminnallisuuden luomiseen myös selainohjelmoinnissa, mutta tähän tarkoitukseen käytetään useammin Javascriptiä. PHP on vapaan lähdekoodin ohjelmointikieli ja sovellusten kehittäminen sen avulla ei tuo lisenssointikuluja. Viralliselta sivustolta löytää kootusti kattavan PHP dokumentaation esimerkkitapauksineen.(PHP.net 2017) PHP valittiin kieleksi tähän työhön, koska sillä

oli vankka tuki niin kehitysympäristön puolesta kuin yleisestikin. Kieli tiedetään toimivaksi ja viralliselta sivustolta löytyy hyvin dokumentoidut käyttöohjeet. Sovelluksen oleellisimmat php-kirjastot olivat Mail.php, libsec.php ja Ssh2_crontab_manager.php. Niillä luotiin ryhtimuistuttajan toiminnallisuus palvelimen puolella. Ssh2_Crontab_manager mahdollisti sähköpostin lähettämisajan ja sähköpostin lähettävän PHP-scriptin kirjaamisen crontabin tiedostotietokantaan. Libsec.php hoiti edellämainitun toiminnan salaamisen. Mail.php sisältää joukon oleellisimpia muuttujia ja metodeja, joiden avulla sähköposti on mahdollista koostaa ja lähettää.

3.2 Bootstrap ja responsive design

Mobiililaitteiden käyttö verkon selailuun on kasvanut ja tulee kasvamaan suurella nopeudella, mutta kuitenkaan suuri osa verkkosivuista ja sovelluksista ei ole optimoitu niitä varten. Mobiililaitteita rajoittaa yleensä näytön koko ja se vaatii erilaisen näkökulman siihen miten sisältö järjestellään ruudulle. Näytön koko vaihtelee laitteesta riippuen, joten on tärkeää että web käyttöliittymä pystyy adaptoitumaan mihin tahansa näytön kokoon(LePage 2017.). Käytännössä se tarkoittaa bootstrapin(http://getbootstrap.com/) tai vastaavan kirjaston ottamista käyttöön, jossa sivun elementit on määritelty järjestäytymään ja skaalautumaan näytön koon ja viereisten elementtien mukaan. Tämä tarkoittaa esimerkiksi otsikkotekstin prosenttuaalisen koon suhdetta näytön kokoon tai otsikkotekstin koon muutosta näytön koon suhteen absoluuttisilla raja-arvoilla ilmaistuna eli tietty otsikkotekstin koko kytkeytyy päälle saavutettaessa tietty näytön pikselikoko.

3.3 Javascript ja jQuery

Javascript on web-sovelluksissa ja selaimessa toimiva ohjelmointikieli, jolla luodaan sivulle toiminnallisuutta. Kaikki suosituimmat verkkoselaimet tukevat Javascriptiä. Javascript kirjastoja löytyy lähes äärettömiin ja niitä on syytä käyttää, jotta voidaan hyödyntää kehittyneempiä, paremmin testattuja ja tilkittyjä ratkaisuja

toiminnallisuuden luomiseen. Tässä projektissa käytettiin neljää erilaista Javascript kirjastoa: jQuery, Parallax, Videojs ja Timepicker.

JQuery tiivistää orginaalin Javascriptin useimmin käytettyjä metodeja omaksi kirjastokseen, joka tarjoaa Javascriptin useimmin tarvittavat toiminnallisuudet kuten DOMin läpikäynnin ja manipuloinnin, tapahtumien hallinnan, animaatiot, ajaxin ja helposti käytettävän APIn, joka toimii useimmilla selaimilla. jQuerya tarvitaan usein muiden Javascript-kirjastojen pohjaksi itse Javascriptin lisäksi ja kyse on hyvin suositusta kirjastosta. Esimerkiksi kaikki sovelluksessamme käytetyt Javascript-kirjastot ja bootstrap mukaan lukien vaativat jQueryn. Juurikin toistuvasti käytetyt Javascript-toiminnallisuudet on valmiiksi tehty ja tiivistetty jQueryssä, joka nopeuttaa ja helpottaa sivuston kehitystyötä. Kirjaston voi löytää osoitteesta:https://jquery.com/.

3.4 Parallax, Video ja Timepicker

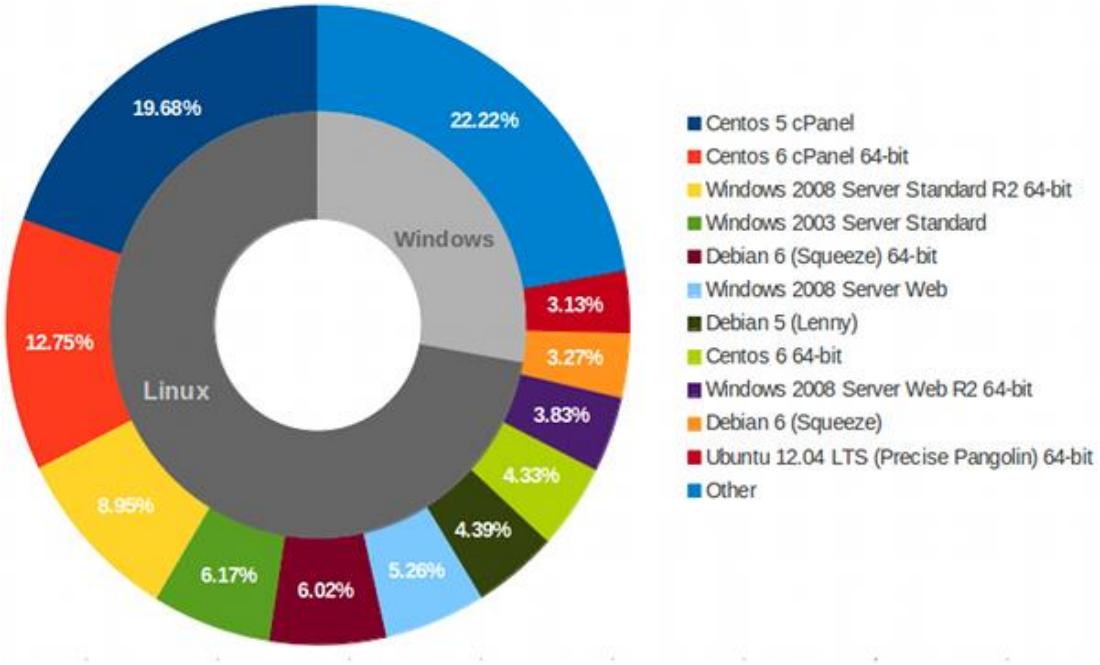
Parallax, Video ja Timepicker olivat jQueryä selvästi pienempiä kirjastoja. Parallax(http://pixelcog.github.io/parallax.js/) valittiin sivustolle saatavan ulkonäön takia, koska se luo trendikkään liikkuvan taustakuvan. Video.js(http://videojs.com/) oli vaihtoehdoista eniten kannatusta ympäri internettiä saanut ja valmis kirjasto video-soittimen rakentamiselle ja sen kaikenlaiselle muokkaamiselle. Video.js:ssä valmiita toimintoja ja ulkoasuja on mielinmäärin. Soitinta voi muokata Javascriptin ja CSS-tyylimäärittelyjen avulla tai liittää jo valmiin teeman ja mukana tulevat toiminnallisuudet. Timepickerin(http://amsul.ca/pickadate.js/date/) avulla luotiin ryhtimuistuttajalle ajan valitsin. Timepickerin kaltaisia kirjastoja ajan tallentamiseen on useita, joista Timepicker valittiin sen yksinkertaisuuden ja tyylikkyyden takia. Näiden kolmen kirjaston valmiit tyylimäärittelyt ja toiminnallisuus tarkoittivat vain yksinkertaisten metodien käyttöä monimutkaisten toimintojen luomiseksi. Jokainen näistä kirjastoista saadaan käyttöön lataamalla kirjasto palvelimelle ja linkittämällä kirjasto etusivun HTML-tiedostoon. Tämän jälkeen kirjaston toiminnallisuudet ovat käytettävissä.

4 KÄYTETTÄVÄT VÄLINEET

4.1 Palvelinympäristö: Bitnami LAMP-stack ja Crontab

Bitnami rakentaa ilmaisia ja valmiiksi konfiguroituja ohjelmistopinoja erilaisiin käyttötarkoituksiin. Käyttötarkoituksina voi olla ohjelmistopinoja verkkokaupoista, moodlen kautta yleiseen palvelinkokoonpanoon kuten LAMP. Kun ohjelmistojen konfiguraatio on suurelta osin kunnossa, kehittäjä voi keskittää aikaansa pinon päälle rakennettavan ohjelmiston kehittämiseen. LAMP-stack on melkein valmiiksi konfiguroitu, koottu ja riippuvuuksista huolehdittu, päivitetty ja helposti asennettava palvelinympäristö, jossa LAMP-kokonaisuuden muodostavat alkukirjainten mukaan: Linux, Apache, MySQL, PHP. Tarkoituksena on asentaa Linux-käyttöjärjestelmään Apache-palvelin, joka tukee MySQL tietokantaa ja PHP-kieltä. LAMP-palvelin voi ottaa erilaisia rooleja WWW-, email- tai tiedostopalvelimena. Tämän kehitysympäristön päälle laitettiin pyörimään Jobergo. Dokumentaatio Bitnamin LAMP-stackin käytöstä on saatavissa osoitteessa: https://docs.bitnami.com/virtual-machine/infrastructure/lamp/

Kehitysalustan pohjimmaisena toimi Linux Mint, joka on yksi suosituimmista Linux-jakeluista. Se valittiin helppokäyttöisyytensä vuoksi, koska siinä kaikki yleisimmät ohjelmistopalikat ovat valmiiksi paikoillaan, kuitenkin taaten helpon pääsyn komentoriville josta kehittyneempi koneen käskytys onnistuu. Mint on Debian-pohjalle rakennettu käyttöjärjestelmä, joka mahdollistaa hyvän yhteensopivuuden monien erilaisten palvelin-sovellusten kanssa, koska Debianilla on pitkä historia tälläisessä käytössä. Eri versiot huomioon ottaen se on myös yksi suosituimmista käyttöjärjestelmistä palvelin-käyttöön kuten alla oleva kuvaaja kertoo.



Kuvio 1. Suosituimmat palvelimen käyttöjärjestelmät(Martinez 2012)

Linux Mintin päälle asennettiin LAMP-stack, joka löytyy osoitteesta: https://bitnami.com/stack/lamp. Asennusvaiheessa määriteltiin yhteysosoitteet, portit ja tunnukset. Näin pienellä toimenpiteellä saatiin Apachen perus-konfiguraatio kuntoon ja palvelin päälle. MySQL-tietokantaa tai PHP-kieltä ei sen kummemmin konfiguroitu, joskin uusimmat versiot molemmista sisältyvät ohjelmistopinoon. Sovellusta kehittäessä ja ajettaessa jouduttiin korjaamaan bugeja ja toimimattomuuksia. Suurta jännitystä liittyi myös tietoturvallisuuteen ja siihen miten haitallinen liikenne pystyttäisiin estämään. Jälkikäteen web-palvelinta säädettäessä suurimmat huolenaiheet liittyivät sisäverkosta internetiin näkyvän järjestelmän kansioiden ja tiedostojen valtuuksiin, joita pyrittiin rajaamaan. Sisäverkkoon ja kiinni web-palvelimeen pääsi reitittimen kautta yhdyskäytäviä pitkin jolloin itse reitittimessä ja Linuxissa olevien palomuurien ip-osoitteiden ja porttien kautta kulkeva liikenne sallittiin vain pakollisen tarpeen mukaan.

4.1.2 Apache server

Apache server asentui melkein valmiiksi konfiguroituna LAMP-stackin mukana. Apache itsessään on yksi suosituimmista web-palvelimista. Sen juuret ovat Unix-pohjaisissa käyttöympäristöissä ja se on myöhemmin käännetty Windowsille ja muille käyttöjärjestelmille. Pitkän historian takia Apachesta löytyy kaikki täyden palvelin-toiminnallisuuden vaatimat komponentit ja se tukee toimintaa laajentavia liitännäisiä. Apache on ilmainen ohjelmisto, jota levittää Apache Software Foundation ja joka on lisensoinut Apachen omalla hyvin vapaalla lisenssillään. Lisenssi sallii lähdekoodin vapaan käytön, kehittämisen ja levittämisen, mutta hyväksyy myös suljettuun järjestelmään koodin kehittämisen.(Lifewire www-sivu 2017.) Itse projektissa Jobergo-sovellus sijoitettiin Apachessa htdocs-juurikansioon, joka oli se piste josta kansio- ja tiedostorakenne alkoi näkyä internettiin päin. Mahdollisten konfiguraatio-ongelmien sattuessa, kuten vaikkapa juurikansion tai yhteysosoitteiden ja porttien uudelleen määrittelyssä, käytettiin pääkansioon sijoitettua htaccess-tiedostoa, joka on palvelimen pää-asetustiedosto. Apachen kotisivut dokumentaatioineen löytyvät osoitteesta: https://httpd.apache.org/.

4.1.3 Crontab

Crontab on työn aikatauluttaja unix-johdannaisiin käyttöjärjestelmiin. Tyypillisesti sillä automatisoidaan järjestelmän ylläpitoa ja hallintaa, vaikka yleisen luonteensa vuoksi se voidaan automatisoida lataamaan tietoja internetistä tai lähettämään sähköpostia (Adminschoice www-sivut). Projektissa käytettiin Linuxin Mintin Crontabia, jolla ajastettiin tehtävien, esimerkiksi scriptien suorittamista. Crontab vaatii ajastukseen viikonpäivän tai päivät ja kellonajan, sekä tiedoston osoitteen tai käskyn joka ajetaan. Erilaisia ajastuksia voi luoda rajattomasti. Ajastukset luodaan Crontabin omilla komennoilla, mutta Jobergo-sovelluksessa käytettiin PHP:lle tehtyä Crontab-kirjastoa, jonka avulla Crontabin ajastuskomennot voidaan antaa suoraan php:lla. Käytännössä Jobergon ryhtimuistuttaja saatiin toimimaan kirjaamalla muistutukset Crontabiin.

4.2 Ohjelmointityökalut: Brackets ja Chrome devtools

Varsinaiseen ohjelmointiin käytettiin Bracketsiä(http://brackets.io/) oli kyseessä sitten PHP, HTML, CSS tai Javascript. Brackets on avoimen lähdekoodin tekstieditori. Brackets itsessään on hyvin kevyt, nopeasti asennettava ja toimiva tekstieditori, johon on saatavilla merkittävästi plug-in tyylisiä lisäosia. Lisäosien ansiosta Bracketsiä pystyy laajentamaan haluamiinsa suuntiin melko kivuttomasti kun haluaa enemmän toiminnallisuutta PHP:n, CSS:n tai Javascriptin koodaamiseen.

Toinen oleellisen tärkeä työkalu verkkosovelluksen kehityksessä oli Chromen Developer Tools. Sen avulla pystyttiin tarkastelemaan ja testaamaan verkkosivun rakennetta. Käytännössä tarkastelu tarkoittaa niin sanottua live-preview tilaa jossa sivu latautuu selaimelle jokaisen muokkauksen jälkeen. Sivun rakennetta pystyy tällöin muokkaamaan, debuggaamaan ja tarkastelemaan reaaliajassa. Tämä tarkoittaa nopeampaa kehitysvauhtia ja helpompaa hienosäätöä pienten osasten suhteen.(Google developer www-sivut 2017) Kuitenkaan tallennus suoraan Developer tools:n kautta varsinaisiin ohjelman tiedostoihin ei onnistunut vaan Bracketsiä piti käyttää välikappaleena. Developer tools tukee vain selainpuolen kieliä kuten sovelluksessa käytetyt HTML, CSS ja Javascript, joten sovelluksen käyttämä PHP ja sen toiminta Javascriptin kanssa ohjelmoitiin Bracketsillä. Developer tools oli erityisen hyvä sovelluksen toimivuutta kokeiltaessa sillä erikokoisia näyttölaitteita pystyttiin mallintamaan, jolloin kaiken kokoiset laitteet saatiin testattua yhdeltä koneelta. Tämän lisäksi Developer tools tarjosi monenlaisia debuggaus-työkaluja prosessien etenemisen ja aikavaativuuksien seurantaan jolloin juuttuneet tai hitaasti toimivat osaset pystyttiin löytämään kätevämmin. Google Developer tools tulee valmiiksi mukana Chrome-selaimessa ja siihen pääsee käsiksi painamalla selaimessa samanaikaisesti Ctrl+Shift+C, jolloin mikä tahansa edessä avoinna oleva sivu aukeaa tarkasteltavaksi.

4.3 GIMP

GIMP eli GNU Image Manipulation Program on avoimen lähdekoodin pohjalle rakentuva monitoiminen ja ilmainen kuvankäsittelyn työkalu. Kuten projektin muissakin työkaluissa, myös tässä laajennettavuutta ja lisäosia on mahdollisuus hyödyntää merkittävissä määrin. GIMPin vaikutus koko projektin kuvalliseen ulkoasuun on merkittävä, koska sen avulla kaikki verkkoon laitetut kuvat luotiin tai muokattiin sopiviksi eli käytännössä kaikki logosta taustakuvaan ja sen elementteihin. GIMPin kotisivut ja dokumentaatio löytyvät osoitteesta: https://www.gimp.org/.

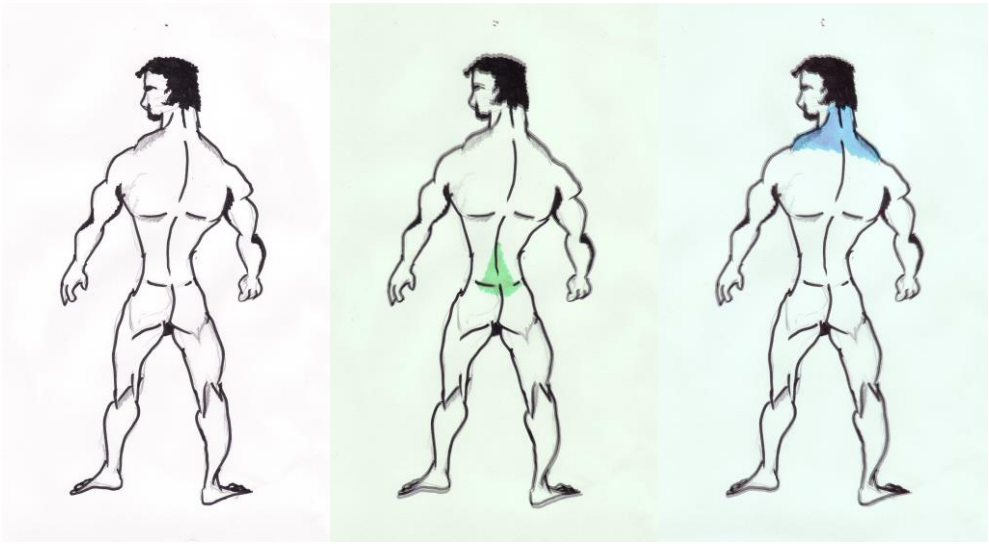
5 SOVELLUKSEN TOTEUTUS

5.1 Jobergon sivupohja

Kun palvelin oli pystyssä ja päivitettynä, aloin etsimään hyvää sivupohjaa jonka päälle pystyi rakentamaan hyvän käyttöliittymän ja sen käyttöliittymän kautta käyttämään haluttuja ohjelmistokomponentteja: ryhtimuistuttajaa ja taukojumpparia. Käyttöliittymän suunnittelun yhteydessä selvitettiin mahdollisuutta kopioida jo toimivia ratkaisuja eli on järkevää etsiä valmis sivupohja työlle, joka on jo todettu hyväksi ja suosituksi. Verkosta löytyi ilmainen vapaasti käytettävä pohja, joka vastasi hyvin sovelluksen tarpeita sopivalla lähdekoodilla, kirjastoilla, ulkonäöllä, responsiivisuudella ja toiminnallisuudella. Koska pohja sisältää Bootstrapin valmiina niin se skaalautuu heti kaikille näyttölaitteille, kuten mobiililaitteille. Lopulta itse pohjasta päädyttiin poistamaan suurin osa sivun sisällöstä ja sisältö korvattiin seuraavaksi kertomillani toteutuksilla. Agency-sivupohjan löytää osoitteesta:https://startbootstrap.com/template-overviews/agency/.

5.2 Visuaaliset palaset: logo, taustakuva ja kipupisteukko

Visuaalinen ilme luotiin GIMP:ä, fysioterapeutin tekemiä piirroksia, ohjelmisto-kirjastoja, omaa taiteellista silmää ja käyttäjäystävällisyyttä yhdistelemällä. Kipupisteukko ja taustakuva saivat alkunsa fysioterapeutin piirtämästä kehon kuvasta, jolla suunnitteluvaiheessa pyrittiin hahmottamaan kipupisteukon muotoa ja toimintaa.



Kuva 1. Kolme eri vedosta kipupisteukosta: vasen, keskimmäinen ja oikea-kuva

Kuvassa numero yksi, ensimmäisenä vasemmalla on perusversio ukosta, josta käsiteltiin GIMP:n avulla viisi muuta versiota. Yksi taustakuvaksi ja neljä muuta tulevat vuorotellen näkyviin käyttäjän valitessa kipupistettä. GIMP:ssä käytettiin vain muutamia oleellisimpia työkaluja, jotka toistuvat myös logon suunnittelussa: paintbrush(“värikynä”), layers(“kerrokset”), bucket fill(“väri-täyttö”), crop(“rajaus”), rotate(“kierrä”) ja scale(“venytä”). Kuvat on jaettu useampaan kerrokseen eli layeriin. Esimerkiksi yllä olevan kuvan numero yksi, kuvasarjan keskimmäinen kuva koostuu kolmesta kerroksesta joista alimmaisena kerroksena on ylhäällä olevasta kuvasarjasta vasemmaisin kuva, jonka päällä on paintbrush-kerros ja näiden päällä on bucketfill-kerros. Paintbrush kerros sisältää vain vihreän jäljen alaselässä. Bucketfill-kerros sisältää taustalla olevan vihreän värin. Jokaista näistä kerrosta pystyy säätämään itsenäisinä palasina, jolloin vaikkapa vihreää taustaa ja sen läpikuultavuutta voi säätää sen vaikuttamatta alempien kerroksien piirtymiseen.

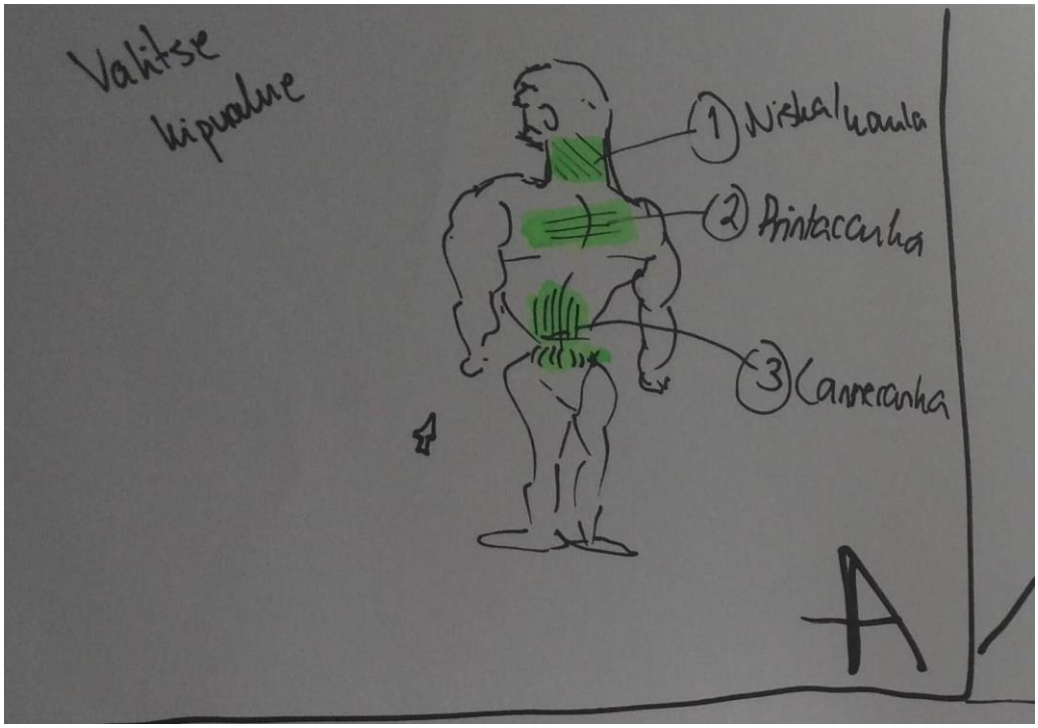
Taustakuvan tapauksessa yhdistettiin kipupisteukon eri kipualueita vastaavat paintbrush-kerrokset yhdeksi kuvaksi. Kun kuvan sisältö oli saatu valmiiksi niin se loppukäsiteltiin, mikä tarkoitti kuvan rajaamista, kiertämällä hienosäätämistä ja venyttämistä. Tässä vaiheessa kuva ladattiin verkkosivulle ja sen sopivuutta sivun kokonaisilmeen kanssa hiottiin askeleittain kunnes kuva oli sopivan kokoinen ja sopivassa kulmassa. Ihan viimeiseksi sivun kuvat pakattiin mahdollisimman pieneen kokoon käyttämällä TinyPNG-palvelua ja ladattiin uudelleen sivulle. TinyPNG löytyy osoitteesta: https://tinypng.com. Esimerkiksi taustakuva vei pakkaamattomana noin 2.5mb ja kun suuri tiedostokoko toistuu kaikkien viiden kuvan kohdalla, se tarkoittaa sivulatausten suurta hidastumista ja käyttömukavuuden heikkenemistä kun sivu ei aukene. Tämä vaikuttaa myös Googlen haussa omaan tulokseen heikentävästi(Hoffman 2013).

5.3 Taustakuvan ja kipupisteukon ohjelmointi

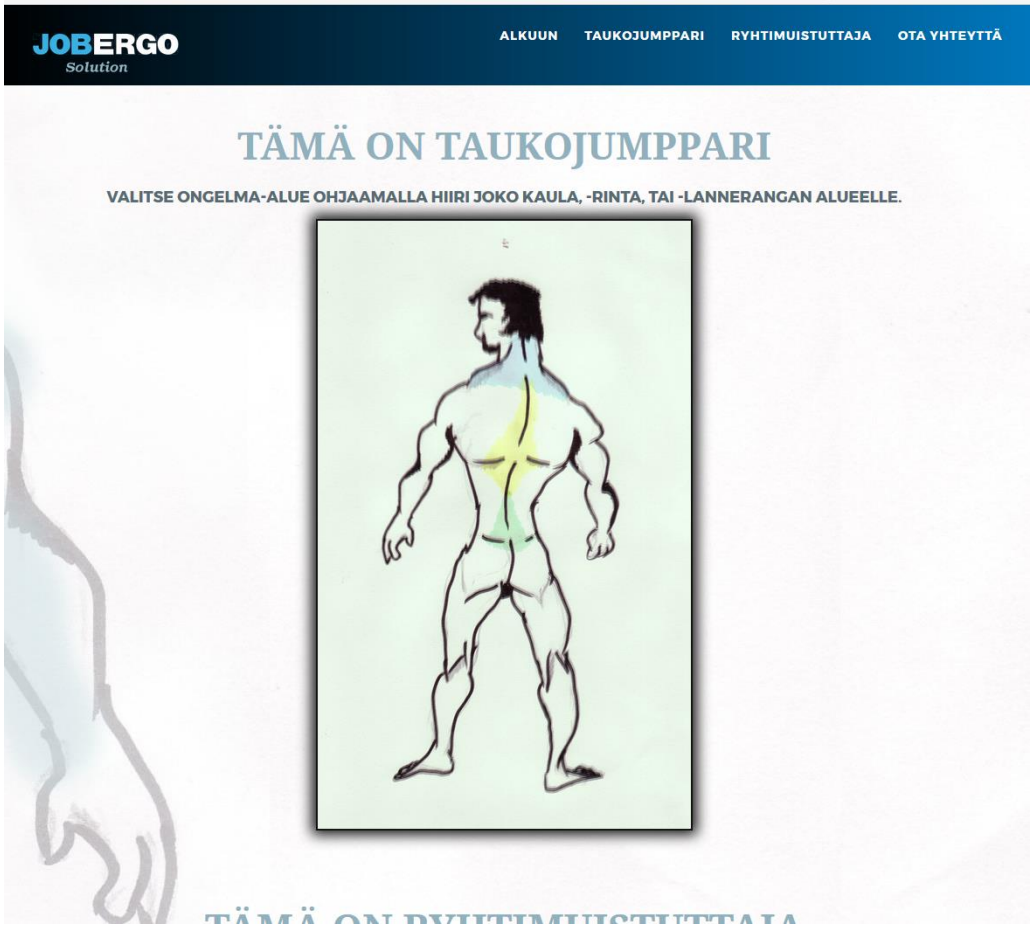
Taustakuvasta ja kipupisteukosta saatiin enemmän hyötyä irti yhdistämällä niihin koodia. Yksi pysyvä trendi internetissä ja verkkosovellusten toteuttamisessa on ollut liikkuvat ja elävät taustakuvat. Tällaista liikkuvaa taustakuvaa kutsutaan termillä: “parallax scrolling”, joka käytännössä toimii siten, että kun käyttäjä liikkuu sivulla ylös tai alaspäin niin taustakuva liikkuu tästä eriävällä nopeudella.(Creativebloq 2017) Tämä luo efektin jossa tausta ja sen päällä olevat elementit kuten teksti, valikot ja kipupisteukko eroavat toisistaan. Käytännössä parallax view oli helppo liittää etusivun HTML-koodin. Sen kirjasto linkitettiin muiden Javascript-kirjastojen mukaan ja käyttöönotto tapahtui lisäämällä etusivun <BODY>-tagin alle <div class="parallax-window"data-parallax="scroll" data-image-src="Materiaali/tausta2.jpg">.

Kipupisteukko toimii pohjimmiltaan valikkona kipupisteen valintaan eli siitä voidaan valita huollettavaksi alueeksi joko kaula-, lanne- tai rinta-ranka. Jokaista valintaa edustaa oma kuvansa ja kun mitään ei ole valittuna, esitetään yleistä kipupistoukon kuvaa jossa näkyvät kaikki valintavaihtoehdot. Siirtämällä hiiri esimerkiksi lannerangan päälle, osoitetaan se valituksi. Tehty valinta linkittää kipupisteukon

ryhtimuistuttajaan, taukojumppariin ja näiden toimintoihin. Ryhtimuistuttajassa valittu kohta merkitsee tiedon siitä osa-alueesta josta käyttäjää muistutetaan sähköpostilla ja taukojumpparissa avautuu lisätietoa tämän kipualueen alta, sekä opetusvideo jonka avulla aluetta voi huoltaa.



Kuva 2. Suunnitteluvaiheen luonnos kipupisteukosta



Kuva 3. Valmis kipupisteukko

Kipupisteukko hyödyntää HTML-kielen harvemmin käytettyä imagemap-toiminnallisuutta, jossa kuvajoukosta luodaan kartta “<map>”-tagilla. Tässä kartassa on kolme sisempää aluetta joita merkitään ”<area>”-tagilla. Jokainen näistä erillisestä alueesta kuvastaa kartalla pinta-alaa jonka päälle hiiren viemällä valikoituu tätä pinta-alaa vastaava kipupiste-alue ja sen kuva, esimerkiksi lanneranka. Kun tämä alue on valittuna niin sen sisällä olevat “<id=”lanneranka”>” ja “<class=”lanneranka”>”-tägit viestivät taukojumpparille ja ryhtimuistuttajalle tehdystä valinnasta. Tämän viestin pohjalta ryhtimuistuttaja tietää mistä alueesta muistutetaan ja taukojumppari tietää mitä aluetta aletaan huoltaa.

```
<map name="image-maps-2016-08-17-125114" id="ImageMapsCom-image-maps-2016-08-17-125114">

<area    name="kaularanka" id="kaularanka"   class="rankateksti"  shape="poly"
coords="1031,836,1079,852,1207,895,1254,841,1115,706,1093,582,1002,537,920,563,897,660,761,711,704,
784,707,834" style="outline:none;" target="_self"    />

<area  name="rintaranka"  id="rintaranka"  class="rankateksti"   alt="" title="" shape="poly"
coords="1113,1090,1081,862,987,845,774,1217,900,1359,960,1366,1032,1264,1089,1220"
style="outline:none;" target="_self"     />

<area name="lanneranka" id="lanneranka"  class="rankateksti"  alt="" title=""  shape="poly"
coords="969,1370,890,1352,785,1591,923,1646,1034,1645,1114,1572" style="outline:none;"
target="_self"     />
</map>
```

Kuva 4. Kipupisteukon toiminnallisuuden luominen

5.4 Taukojumpparin rakentaminen

Taukojumppari on toinen kahdesta sovelluksen pääkomponentista. Sen tehtävä on näyttää käyttäjälle kehonhuollossa auttavaa tietoa ja kehonhuoltovideo. Kipupisteukosta kipualeen valinnan jälkeen käyttäjälle esitetään lisätietoa kipualueella yleisesti esiintyvistä ongelmista ja annetaan kaksi vaihtoehtoa, joko hyppääminen suoraan ryhtimuistuttajaan tai siirtyminen kehonhuolto-videoon. Lisätieto-palsta ja video-opastus on molemmat ohjelmoitu yhdistäen HTML, CSS ja Javascript-koodia. Lisätieto-palsta toimii omilla uniikeilla tägeillaan(“<la1>”,”<la2>”,”<la3>”), jotka on Javascriptin avulla piilotettu käyttöliittymässä. Jokainen tägi sisältää id:n, joka linkittää valitun kipualueen ja uniikin tägin. Javascript tekee id:n avulla uniikin tägin ja sen sisältämät rakenteet näkyviksi kun käyttäjä valitsee kipualueen. Tämä rakenne on HTML-koodin “<ul>” ja “<li>”-tägien muodostama lista kipualueesta ja siihen liittyvistä yleisimmistä oireista. Samalla Javascript pitää muut kipualueet piilossa komennolla: “$("uniikki_tägi").hide();”. Listan näyttäminen taas tapahtuu komennolla: ”$("uniikki_tägi").show();”.

Kehonhuoltovideon katseluun pääsee klikkaamalla opi ryhtijumppa -painiketta. Nappulaa painaessa videosoitin alustetaan ja sille asetetaan alkuarvoiksi palvelimelta löytyvä aiemmin valitun kipualueen video. Kipupisteukko piilotetaan ja sen tilalla näytetään videosoitinta. Lopuksi soitin saa videon käynnistävän komennon: “player.play();”. Videosoitin itsessään on jatkuvasti upotettuna, mutta piilotettuna sovelluksen HTML-koodiin.

5.5 Ryhtimuistuttajan rakentaminen

Ryhtimuistuttajan tehtävänä on kerätä käyttäjältä tarvittavat tiedot ja muistuttaa kerättyjen tietojen avulla sähköpostia lähettäen käyttäjää kehonhuollosta. Sähköpostimuistutuksen tullessa Android-puhelimeen ja Googlen gmail-sovellukseen se antaa käyttäjälle hälytysmerkin. Hälytin päätettiin toteuttaa näin, koska lähestymistapa teki siitä yleisesti hyödynnettävän ja riippumattoman paikasta, sillä älypuhelimia alkaa olla jo useimmilla ja usein kyseessä on Android, johon linkittyy Google-tili(Wearesocial www-sivut, 2017).

Ryhtimuistuttaja kerää käyttäjältä tiedot html-lomakkeeseen, josta parsitaan muistutukseen tarvittavat tiedot: kipualue, viikonpäivät, kellonaika ja sähköpostiosoite. Kipualue saadaan kipupisteukolta. Viikonpäivän valinta toteutettiin checkbox-tyyliin ja kellonajan valinnassa hyödynnettiin Javascriptin Timepicker-kirjastoa. Kun tiedot on syötetty, käyttäjä aloittaa muistutuksen painamalla: “Aloita muistutus”-painiketta, joka kutsuu palvelimelta PHP:lla luodun parsija-operaation. Parsija asettaa viikonpäivät ja kellonajan Crontabiin ja luo käyttäjän sähköpostiosoitteen nimisen tiedoston, jonka Crontab käynnistää valittuna aikana. Käynnistyessään käyttäjän sähköpostiosoitteen mukaan nimetty tiedosto ottaa yhteyden Googlen sähköpostipalvelimelle, jonka kautta tämän tiedoston sisältämä muistutusviesti lähetetään. Käyttäjä kuulee puhelimessaan äänimerkin ja voi avata linkin kuntoutus-videoon.

Parsija koostaa siis käyttäjän sähköpostin mukaan nimetyn tiedoston ja tallentaa muistutettavan ajan ja tämän sähköposti-tiedoston tiedostopolun Crontabiin. Muistutettavan ajan Crontabiin voi kirjata 'Ssh2_crontab_manager.php' nimisen kirjaston ja sen vaatiman komentoyhteyden salaukseen erikoistuneen 'phpseclib.php'-kirjaston avulla. Näiden avulla palvelun käyttäjän antama ”Aloita muistutus”-komento muodostaa palvelimeen salatun ssh-yhteyden PHP-koodista käsin, kirjautuu palvelimelle ja kirjoittaa käyttäjän lomakkeeseen antamat tiedot Crontabiin. Tämä on lopulta varsin lyhyt operaatio. Aluksi PHP-koodissa annetaan oman palvelimen ip-

osoite, portti ja kirjautumistunnukset. Sitten annetaan kellonaika, päivät ja tiedostopolku käynnistettävään sähköposti-tiedostoon. Tämän operaation voi nähdä alta.

```
$crontab = new Ssh2_crontab_manager('83.102.66.38', '22', 'Käyttäjätunnus', 'Salasana');
$crontab->append_cronjob(''.$kellonaika.' * * '.$paivat.'  curl http://localhost/log/email'.$sahkoposti.'.php');
```

Kuva 5. Ryhtimuistuttajan ajastaminen

Parsija koostaa myös yllä olevassa kuvassa numero viisi esiintyvän “email.’$sahkoposti.’.php”-tiedoston, joka sisältää käyttäjälle lähtevän sähköpostin ja jonka Crontab käynnistää muistutuksen ajankohtana. Sähköposti on kirjoitettu PHP:lla Mail.php kirjastoa apuna käyttäen. Lähtevä sähköposti tarvitsee lähettäjän ja vastaanottajan osoitteen, Googlen sähköpostipalvelimen osoitteen ja sen tunnukset. Lisäksi viestikenttään lisätään linkki kipupisteukosta valittuun videoon. Lähettäjän osoite on sovelluksen kotiosoite: jobergosolution@gmail.com, joka toimii myös käyttäjätunnuksena otettaessa yhteyttä Googlen sähköpostipalvelimeen smtp-protokollan avulla. Tämän lisäksi tunnistautuminen vaatii salasanan. Käyttäjän sähköpostiosoite saadaan lomakkeesta. Lähetettävä viesti kirjoitettiin PHP:n sisään viestikenttään HTML-kieltä käyttäen. Ainut muuttuva osuus viestikentässä on linkki joka ohjaa johonkin kolmesta eri kipualueen videosta. Alla olevassa kuvassa 6, esitetään parsijan toteutus. Lopuksi parsija tallentaa käyttäjän sähköpostiosoitteen nimisen tiedoston palvelimelle.

```
require_once "Mail.php";

$from = "jobergosolution@gmail.com";
$to = '.$sahkopostiosoite.';
$host = "ssl://smtp.gmail.com";
$username = "jobergosolution@gmail.com";
$password = "Salasana";

$message .= "<tr><td><strong>VIDEOLINKKI (main):</strong> </td><td> </td></tr>";

$headers .= "MIME-Version: 1.0\r\n";
$headers .= "Content-Type: text/html; charset=ISO-8859-1\r\n";
$subject = "Klikkaa tästä ja harjoitellaan - ryhtimuistuttaja";
$body = "Klikkaa tästä ja ryhtiharjoitteeseen  http://jobergo.tk/'.$Valittuongelmaalue.'.html

Jos haluat lopettaa muistuttajan niin vastaa tähän sähköpostiin: haluan lopettaa muistutuksen.
";
```

Käyttäjän sähköpostiosoite

Googlen sähköpostivalvelimelle kirjautuminen

Linkin luonti ongelma-alueeseen

Kuva 6. Käyttäjän sähköposti-nimisen tiedoston parsiminen

Viimeisenä asiana oli käyttäjän puhelimeen pirahtava muistutus. Sovellus lähettää käyttäjälle sähköpostin haluttuna aikana. Kätevintä oli käyttää valmista usein käytettyä ohjelmistoa käyttäjän laitteessa toimivan muistuttajan pohjana, jolloin saadaan tavoitettua mahdollisimman laaja kohderyhmä käyttäjiä ja käyttäjän laitteessa toimivan muistuttajan luomiseen ei mene huomattavia resursseja. Pohjana toimi Androidille oleva Gmail-sovellus, jolla saadaan melkein kaikki älypuhelimet katettua ja muistuttaja on aina mukana mikäli Gmail-sovellus ja sähköposti löytyvät käyttäjältä. Gmail-sovellus saadaan piippaamaan kun asetetaan tämän ohjelman asetuksista Jobergon sähköposti-osoitteesta tulevat viestit tärkeiksi ja varmistetaan että tärkeistä viesteistä ilmoittaminen on äänellisellä. Tällöin kun viesti saapuu puhelimeen se pitää merkkiäänen ja siirtyy älypuhelimen ilmoituksiin(notification). Sieltä viestin avaamalla pääsee linkkiä pitkin sivulle, joka toistaa ryhtiharjoitteen videon. Gmail-sovellus Androidille löytyy osoitteesta:https://play.google.com/store/apps/details?id=com.google.android.gm&hl=fi. Tämä oli ryhtimuistuttajan toteutus.

6 JÄLKIPOHDINTA

Kokonaisuudessaan matka opinnäytetyön aloituksesta tähän pisteeseen oli pitkä. Se opetti paljon itselle niin opiskelemastani alasta ja myös sinnikkyydestä tehdä asiat loppuun vaikka ne kuinka venyisivät, enkä koskaan enää halua venyttää asioita tällä tavalla. Jättimäinen apina selästäni on hypännyt pois. Mielestäni opinnäytetyö kuvastaa hyvin suhteellisen laajaa orastavaa osaamistani full-stack koodaajasta, palvelin puolen säätäjään ja hivenen taiteelliseen käyttöliittymän tekijään. Yhteistyö Fysioterapeutin kanssa ja se että yrittää tehdä hyödyllistä sovellusta on kuitenkin arvokasta kokemusta. Itse Jobergo-sovellus ei elänyt lopulta sen pidempään, mutta se toimii hyvänä näytetyönä osaamisestani ja on toistaiseksi löydettävissä osoitteesta: http://jobergo.jyrin.com/. Näihin sanoihin, adiós!

LÄHTEET

Adminschoice www-sivut. Viitattu 18.7.2017.
http://www.adminschoice.com/crontab-quick-reference

Bourque, P & Fairley, R. 2014. The Software Engineering Body of Knowledge (SWEBOK). Viitattu 15.11.2017.
https://www.computer.org/web/swebok/v3;jsessionid=306b2a8638c35ab316ee257d69de

Creativebloq www-sivut. 2017. Viitattu 29.10.2017.
http://www.creativebloq.com/web-design/parallax-scrolling-1131762
Ergopro www-sivut. Viitattu 1.6.2017. http://www.ergopro.fi/

Google developers www-sivut. 2017. Viitattu 20.7.2017.
https://developers.google.com/web/tools/chrome-devtools/

Lepage, P. 2017. Responsive Web Design Basics. Viitattu 4.6.2017.
https://developers.google.com/web/fundamentals/design-and-ui/responsive/

Lifewire www-sivut. 2017. Viitattu 17.7.2017. https://www.lifewire.com/definition-of-apache-816509

Martinez J. 2012. What is the best server operating system?
http://www.memset.com/blog/which-are-most-popular-opearting-systems-our-cloud/

Moz.com www-sivut. 2013. Viitattu 10.10.2017. https://moz.com/blog/how-website-speed-actually-impacts-search-ranking

Office ergonomics software www-sivut. Viitattu 1.6.2017. https://www.enviance.com/office-ergonomics-software-trial-video

PHP.net www-sivut. Viitattu 1.4.2017. http://www.php.net/

Seppälä, N. 2016. Jobergo hyvinvointisovellus: Motorisen kontrollin harjoitteiden käytettävyystestaus. Opinnäytetyö. Satakunnan ammattikorkeakoulu. Viitattu 19.5.2017. https://www.theseus.fi/handle/10024/118363

Valade, J. 2011. Storing data with php — flat file or database?. Viitattu 7.6.2017 http://www.dummies.com/programming/databases/storing-data-with-php-flat-file-or-database/

Wearesocial www-sivut. 2017. Viitattu 25.11.2017. https://wearesocial.com/special-reports/digital-in-2017-global-overview