



**WEB-SOVELLUS
FYSIOTERAPEUTTISTEN
HARJOITUSOHJELMIEN
LUOMISEEN JA SUORITTAMISEEN**

Tomi Leinonen

Opinnäytetyö
Toukokuu 2015
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

TOMI LEINONEN:

Web-sovellus fysioterapeuttisten harjoitusohjelmien luomiseen ja suorittamiseen

Opinnäytetyö 30 sivua
Toukokuu 2015

Tämän opinnäytetyön tarkoituksena oli suunnitella ja kehittää web-sovellus, jonka avulla fysioterapeutti pystyisi luomaan asiakkailleen ohjeistettuja harjoitusohjelmia. Työn tarkoituksena oli myös kehittää tekijänsä kykyä web-ohjelmoinnissa ja käyttöliittymäsuunnittelussa.

Työtä varten tarkasteltiin joitakin samankaltaisia olemassa olevia sovelluksia ja niiden ominaisuuksia. Tässä yhteydessä huomattiin, ettei sovellusta joka mahdollistaisi helpon yksilöllisten harjoitusohjelmien luomisen ja suorittamisen näyttäisi olevan.

Sovellukselle suunniteltiin toteutettavaksi kaksi erilaista käyttöliittymäkokonaisuutta, yksi fysioterapeuteille harjoitusohjelmien luontiin ja toinen heidän asiakkailleen harjoitusten suorittamiseen. Asiakkaiden käyttöliittymästä pyrittiin tekemään toimiva myös mobiililaitteilla. Sovellukseen suunniteltiin myös tietokanta tarpeellisten tietojen tallentamiseen.

Sovelluksesta luotiin prototyyppiversio, jonka teknisen toteutuksen yksityiskohdat on kuvattu tässä raportissa. Toteutuksessa käytettiin HTML-, JavaScript- ja PHP-ohjelmointikieliä. Tietokantana käytettiin MySQL-kantaa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Computer Science
Software Engineering

TOMI LEINONEN:

Web Application for Creating and Performing Physiotherapeutic Exercise Sets

Bachelor's thesis 30 pages

May 2015

The purpose of this thesis was to design a web application that could be used by physiotherapists to create exercise sets for their clients. Increasing skills in web development and user interface design for the author of this thesis was also intended.

In order to design the application, some similar existing applications were searched for and their features were researched. The main point found was that there didn't seem to exist an application that would allow for both creation and display of highly personalized exercise sets.

The application was designed to contain two different user interfaces: one for creation of exercise sets by physiotherapists, the other to assist their clients in performing the sets. The intent was for the clients' interface to be functional on both desktops and mobile devices. A database was also designed for storing any necessary information.

A prototype of the application was completed and its technical details are covered in this report. The programming languages used in development were HTML, JavaScript and PHP. MySQL was used as the database.

Key words: web development, JavaScript, PHP, HTML, physiotherapy

SISÄLLYS

1	JOHDANTO.....	6
2	MUITA SAMANKALTAISIA SOVELLUKSIA.....	7
3	KÄYTTÖTARKOITUS JA OMINAISUUDET	8
3.1	Tarkoitus	8
3.1.1	Hyödyt.....	8
3.1.2	Erot muihin sovelluksiin	8
3.2	Ominaisuudet	9
3.2.1	Harjoitusten esittäminen käyttäjälle.....	9
3.2.2	Harjoitusten luominen.....	9
3.2.3	Harjoitusohjelmien luominen ja aikataulutus	9
3.2.4	Käyttäjien hallinta	10
4	OHJELMISTOSSA KÄYTETTÄVÄT ALUSTAT JA TEKNIIKAT	11
4.1	HTML-kieli.....	11
4.2	CSS-tyylikieli.....	11
4.3	JavaScript-kieli ja jQuery-kirjasto	11
4.4	Bootstrap-webohjelmistokehys.....	12
4.5	PHP-kieli.....	12
4.6	MySQL-tietokanta ja SQL-kyselykieli.....	13
5	OHJELMISTON SUUNNITTELU.....	14
5.1	Käyttöliittymä	14
5.1.1	Kalenterinäkymä	14
5.1.2	Harjoitusnäkymä	15
5.1.3	Harjoitusten ja harjoitusohjelmien luomisnäkymät	16
5.1.4	Harjoitusohjelmien ajastus ja käyttäjähallintanäkymä.....	18
5.2	Tietokanta	19
6	TEKNINEN TOTEUTUS	21
6.1	Web-käyttöliittymien toteutus	21
6.1.1	Sisäänkirjautumissivu	21
6.1.2	Siirtyminen fysioterapeutin käyttöliittymänäkymien välillä.....	22
6.1.3	Harjoitusten ja harjoitusohjelmien luomisnäkymien toiminnot.....	22
6.1.4	Käyttäjähallinnan ja käyttäjän kalenterinäkymän toiminnot.....	24
6.1.5	Harjoitusten esitysnäkymän toiminnot.....	26
6.2	Tietokantayhteyksien toteutus	28
7	YHTEENVETO	29
	LÄHTEET.....	30

LYHENTEET JA TERMIT

Android	Käyttöjärjestelmä, jota käytetään pääasiassa mobiililaitteissa.
AJAX	Asynchronous JavaScript and XML. Yleisnimi tekniikoille joilla web-sivu kommunikoi serverin kanssa asynkronisesti.
Bootstrap	Erityisesti responsiivisten web-sivujen toteuttamiseen soveltuva web-ohjelmistokehys.
CSS	Cascading Style Sheets. Kieli jolla määritellään web-sivustoille tyyliohjeita
ER-kaavio	Entity-Relationship kaavio. Relaatiotietokantojen rakenteen kuvaamiseen käytetty kaaviomalli.
HTML	HyperText Markup Language. Web-sivujen luomiseen käytetty ohjelmointikieli.
JavaScript	Web-ohjelmoinnissa käytetty ohjelmointikieli.
jQuery	Web-käyttöliittymien ohjelmoinnissa auttava JavaScript-aliohjelmakirjasto.
JSON	JavaScript Object Notation. JavaScript-olioiden muodostamiseen käytetty tiedostomuoto.
Liitostaulu	Relaatiotietokannassa oleva tietotaulu, jolla voidaan määrittellä kahden eri taulun väliset yhteydet.
PHP	PHP- Hypertext Preprocessor. Pääasiassa web-ohjelmointiin käytetty ohjelmointikieli.
Relaatiotietokanta	Tietokanta, jossa voidaan määrittellä yksittäisten tietojen väliset yhteydet.
Responsiivisuus	Sovelluksen kyky mukautua automaattisesti erikokoisille laitteille.
Sovelluksen sisäinen osto	Lisäsisällön ostaminen sovellukseen käyttäen sovellukseen sisäänrakennettua ominaisuutta.
SQL	Structured Query Language. Tietokantakyselyjen tekemiseen käytetty kieli.
W3C	World Wide Web Consortium. Monia web-standardeja ylläpitävä yritysten ja yhteisöjen yhteenliittymä.
WHATWG	Web Hypertext Application Technology Working Group. HTML-standardeja kehittävä ja ylläpitävä yhteisö.

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on kehittää fysioterapeuttien harjoitusten ohjauksessa tekemisessä avustava web-sovellus, jonka avulla fysioterapeutti pystyy luomaan yksilöityjä harjoitusohjelmia asiakkailleen ja joka mahdollistaa näiden harjoitusohjelmien helpon ohjeistetun suorittamisen. Sovelluksesta pyritään luomaan mahdollisimman käyttäjäystävällinen ja sen käyttö pyritään mahdollistamaan myös mobiililaitteilla.

Opinnäytetyön oppimistavoitteena on kehittää tekijänsä kykyä suunnitella ja toteuttaa ohjelmisto kokonaisuudessaan, yleistä- ja erityisesti web-käyttöliittymien suunnittelu-kykyä sekä syventää osaamista web-ohjelmoinnista ja muista siihen liittyvistä tekniikoista.

Tässä raportissa esitellään sovelluksen ominaisuudet ja tekniset yksityiskohdat, sekä käydään läpi sovelluksen suunnittelun ja toteutuksen vaiheet. Raportin toisessa luvussa esitellään joitakin samankaltaisia olemassa olevia sovelluksia ja tarkastellaan niiden ominaisuuksia. Kolmannessa luvussa käsitellään tarkemmin sovelluksen käyttötarkoitusta, ominaisuuksia ja mahdollisia hyötyjä. Neljännessä luvussa esitellään sovelluksen teossa käytetyt tekniikat, alustat ja ohjelmointikielet yleisellä tasolla ja kerrotaan, miksi niitä on päätetty käyttää tässä työssä. Viidennessä ja kuudennessa luvussa kerrotaan yksityiskohtaisemmin sovelluksen suunnittelusta ja toteutuksesta.

2 MUTTA SAMANKALTAISIA SOVELLUKSIA

Suunnittelua varten tarkasteltiin joitakin samankaltaisia olemassa olevia sovelluksia. Sovelluksia tarkasteltaessa etsittiin erityisesti tämän työn kannalta hyödyllisiä ominaisuuksia, sekä paremmin toteutettavissa olevia ideoita.

Päivittäiset Treenit, Daily Workout Apps, LLC

Päivittäiset Treenit on Android- sovellus, joka sisältää joitakin ohjevideoilla esiteltäviä harjoitusohjelmia. Käyttäjä voi valita ohjelmien keston ja niitä on mahdollista hankkia lisää sovelluksen sisäisillä ostoilla. Sovelluksessa ei kuitenkaan ole mahdollisuutta harjoitusohjelmien luontiin tai muokkaukseen. (Päivittäiset Treenit 2015.)

Training Apps, Training Apps Finland Oy

Training Apps on Android- sovellus, joka tarjoaa valmiita, lähinnä kunnon kohottamiseen keskittyviä, harjoitusohjelmia. Sovellus sisältää harjoitusten ajastamiseen tarkoitettua päiväkirjaominaisuuden ja testejä kehittymisen seurantaan, mutta ei minkäänlaista mahdollisuutta yksilöllisten harjoitusohjelmien luomiseen. (Training Apps 2015.)

Workout Trainer, Skimble Inc.

Workout Trainer on Android- sovellus, jossa käyttäjä voi valita useista valmiista harjoitusohjelmista tai koostaa omansa tavoitteidensa pohjalta sovelluksen apuominaisuutta käyttäen. Harjoituksia esitetään video-, kuva- ja ääniohjeiden kanssa. Itse luotuja harjoitusohjelmia voidaan jakaa muiden käyttäjien kanssa sähköpostin, Facebookin tai Twitterin välityksellä. (Workout Trainer 2015.)

PhysioTools, PhysioTools Oy

PhysioTools on Windowsilla tai suoraan web-selaimessa käytettävissä oleva ohjelmisto, jolla voidaan koostaa harjoitusohjelmia ohjelmiston sisältämistä kuvista, videoista ja ohjeteksteistä. Ohjelmisto mahdollistaa näiden harjoitusohjelmien tulostamisen tai lähettämisen sähköpostilla asiakkaille. Ohjelmisto on kuitenkin tarkoitettu vain harjoitusohjelmien luomiseen, eikä siinä ole erillisiä ominaisuuksia harjoituksia tekeville asiakkaille. (PhysioTools 2015.)

3 KÄYTTÖTARKOITUS JA OMINAISUUDET

3.1 Tarkoitus

Työssä tehtävää sovellusta käytetään avustamaan fysioterapeuttisten harjoitusohjelmien luomisessa ja suorittamisessa. Sovellus on tarkoitettu pääasiassa fysioterapeuteille ja heidän asiakkailleen.

3.1.1 Hyödyt

Sovelluksen pääasiallinen hyöty on harjoitusohjelmien luomisen ja suorittamisen helpottaminen. Fysioterapeutti pystyy määrittämään asiakkailleen yksilölliset harjoitusohjelmat ja asettamaan ne tehtäväksi haluttuina aikoina ilman että asiakkaalle täytyy tehdä paperisia ohjeita tai aikatauluja. Harjoitusohjelmien suorittaminen helpottuu, kun harjoituksia tehtäessä käyttäjällä on jokaista harjoitusta varten ohjevideo, ilman että käyttäjän täytyy itse hakea haluamansa video. Koska sovellus on käytettävissä myös mobiililaitteilla, käyttäjä voi helposti tehdä harjoituksiaan haluamassaan paikassa.

3.1.2 Erot muihin sovelluksiin

Aiemmin mainituista sovelluksista voidaan havaita, että ratkaisuja valmiiden harjoitusohjelmien ohjeistettuun suorittamiseen ja omien harjoitusohjelmien luontiin on olemassa. Erityisesti mobiilisovelluksia tähän tarkoitukseen vaikuttaisi olevan monia. Kuitenkaan ratkaisua joka mahdollistaisi sekä fysioterapeutin asiakkailleen laatimat yksilölliset harjoitusohjelmat, että näiden ohjelmien ohjeistetun suorittamisen sovelluksen avulla, ei näyttäisi löytyvän. Tässä työssä tehtävään sovellukseen pyritään luomaan juuri tällaisen yksilöllisen ohjaamisen mahdollistavat ominaisuudet.

3.2 Ominaisuudet

Sovellukseen tullaan toteuttamaan erilliset käyttöliittymäkokonaisuudet fysioterapeuteille ja heidän asiakkailleen. Näihin käyttöliittymiin pyritään luomaan tässä kappaleessa esiteltävät ominaisuudet.

3.2.1 Harjoitusten esittäminen käyttäjälle

Käyttäjä voi suorittaa harjoitusohjelmia valitsemalla hänelle määritellyn ohjelman sovelluksen kalenterinäköymästä. Käyttäjälle esitetään harjoitusohjelmassa määritellyt harjoitukset ohjevideoiden ja tekstien avulla. Sovellus siirtyy harjoituksissa eteenpäin joko käyttäjän painaessa ruudulla näkyvää nappia tai automaattisesti ajastettuna, riippuen harjoituksesta. Kun harjoitusohjelma on tehty loppuun, se merkitään käyttäjän kalenteriin suoritetuksi.

3.2.2 Harjoitusten luominen

Fysioterapeutti voi luoda sovelluksessa uusia harjoituksia käytettäväksi harjoitusohjelmissa. Harjoitus luodaan asettamalla harjoitukselle ohjetekstit, ajastusasetukset ja harjoituksen yhteydessä esitettävät videot. Harjoituksille voi myös määrittää erillisen ohjevaiheen, johon voidaan asettaa varsinaisesta suoritusvaiheesta erillinen video.

Fysioterapeutti voi määrittää luomansa harjoituksen yksityiseksi, jolloin se ei ole muiden sovellusta käyttävien fysioterapeuttien käytettävissä.

3.2.3 Harjoitusohjelmien luominen ja aikataulutus

Sovellukseen tehdään erillinen käyttöliittymä, jolla fysioterapeutti voi luoda harjoitusohjelmia. Tässä käyttöliittymässä voidaan koostaa harjoitusohjelma yksittäisistä harjoituksista sekä muokata harjoitusten ohjeita ja ajastusasetuksia harjoitusohjelmakohtaisesti. Harjoitusohjelmissa voidaan käyttää sekä sovelluksessa valmiina olevia että fysioterapeutin itsensä tai muiden sovellusta käyttävien fysioterapeuttien luomia harjoituksia.

Fysioterapeutilla on käytettävissä kalenterinäkyvä josta hän voi asettaa luomansa harjoitusohjelmat suoritettavaksi rekisteröidyille käyttäjilleen haluttuna päivämääränä.

3.2.4 Käyttäjien hallinta

Käyttäjien lisääminen sovellukseen tapahtuu fysioterapeuttien toimesta. Käyttäjälle määritellään nimi ja sähköpostiosoite, jolloin sovellus luo käyttäjälle salasanan ja lähettää sen automaattisesti annettuun sähköpostiosoitteeseen. Fysioterapeutille annetaan mahdollisuus hallinnoida kaikkien luomiensa käyttäjien tietoja ja lisätä näille harjoitusohjelmia.

4 OHJELMISTOSSA KÄYTETTÄVÄT ALUSTAT JA TEKNIIKAT

4.1 HTML-kieli

HTML on merkintäkieli, jota käytetään web-sivustojen luomiseen. HTML-koodi muodostuu erilaisilla tunnisteilla merkityistä HTML-elementeistä ja niiden sisällöistä, jotka web-selain osaa tulkita ja muuttaa käyttäjälle näkyväksi web-sivuksi. HTML-elementtejä ovat esimerkiksi tekstit ja niiden muotoilut, kuvat sekä linkit. (Mozilla Developer Network 2015a.)

HTML-kielen vuonna 1989 alkanut kehitys jatkuu edelleen. Kielen nykyinen, vuonna 2014 virallisesti julkaistu, HTML5-versio lisää uusia, erityisesti median toistoon ja monimutkaisempien web-sovellusten kehitykseen tarkoitettuja elementtejä ja ominaisuuksia. HTML5 pyrkii myös parantamaan web-sovellusten toimivuutta mobiililaitteilla. HTML-kieltä ja -standardia ylläpitävät ja kehittävät W3C- ja WHATWG-yhteenliittymät. (W3C 2014.)

4.2 CSS-tyylikieli

CSS on pääasiassa web-dokumenttien muotoilun ja tyylin hallintaan tarkoitettu kieli. CSS-kielellä määritellään dokumenteille tyyllisääntöjä, joita web-selain noudattaa piirtäessään HTML-dokumentin käyttäjälle näkyväksi. Tyyllisäännöillä voidaan ohjata esimerkiksi elementtien kokoa ja asettelua sekä tekstien muotoilua. Säännöt asetetaan yleensä koskemaan tiettyä HTML-elementtien luokkaa, jolloin kaikille kyseisen luokan elementeille sivustolla voidaan helposti antaa yhtenäinen tyyli. (WebPlatform.org 2015.)

4.3 JavaScript-kieli ja jQuery-kirjasto

JavaScript on tulkittava komentosarjakieli jota käytetään pääasiassa web-sivujen sisällön dynaamiseen luontiin, muokkaukseen ja toiminnallisuuden toteuttamiseen. Web-selain tulkaa JavaScript-koodin ja asettaa sen määrittelemän toiminnallisuuden näyt-

tämälleen web-sivulle. Käytännössä kaikki yleisimmät selaimet tukevat JavaScriptiä. (Mozilla Developer Network 2015b.)

JavaScriptillä voidaan muuttaa HTML-elementtien sisältöä sekä kommunikoida asynkronisesti web-palvelimen kanssa (nk. AJAX-tekniikat), ilman että selaimessa näkyvää sivua on tarpeellista ladata uudelleen. Tämä tekee JavaScriptistä sopivan kielen web-selaimessa toimivien sovellusten toteutukseen. Tässä työssä JavaScriptiä tullaankin käyttämään käyttöliittymän toiminnallisen logiikan toteutukseen.

JavaScriptiin on tarjolla jQuery-kirjasto, joka yksinkertaistaa useiden JavaScriptin toimintojen käyttöä. jQueryn avulla voidaan viitata HTML-elementteihin, muokata niiden ominaisuuksia ja käyttää AJAX-tekniikoita huomattavasti pienemmällä määrällä koodia kuin pelkillä JavaScriptin perustoiminnoilla. jQuery helpottaa myös funktioiden asettamista suorittumaan tiettyjen tapahtumien, kuten HTML-dokumentin latautumisen, yhteydessä. (Murphey 2012.)

4.4 Bootstrap-webohjelmistokehys

Bootstrap on HTML-, CSS- ja JavaScript-ohjelmistokehys, jota käytetään web-sivustojen kehittämiseen. Bootstrapiin sisältyy huomattava määrä valmiita malleja ja elementtejä web-käyttöliittymien rakentamiseen ja asetteluun. Kaikista näistä malleista voidaan helposti tehdä responsiivisia, eli automaattisesti erikokoisille laitteille mukautuvia. (w3schools.com 2015.)

Koska tässä työssä tehtävän sovelluksen on tarkoitus toimia osittain myös mobiililaitteilla, on Bootstrapin kaltaisen, responsiivista suunnittelua tukevan ohjelmistokehityksen käyttö erittäin hyödyllistä.

4.5 PHP-kieli

PHP on erityisesti web-kehityksessä käytetty, suoritusvaiheessa tulkittava ohjelmointikieli. Toisin kuin esimerkiksi HTML tai JavaScript, PHP-koodi suoritetaan täysin web-palvelimella, joka lähettää vain PHP-koodin luoman HTML-koodin tai muun datan

käyttäjän selaimelle. PHP:tä kehitetään avoimen lähdekoodin mallilla ja sen käyttö on ilmaista. (Olson 2015.)

Tässä työssä PHP:tä tullaan käyttämään lähinnä sovelluksen ja tietokannan välisen rajapinnan toteuttamiseen. PHP soveltuu tähän tarkoitukseen hyvin, koska se sisältää laajan MySQL-tietokantojen kanssa kommunikointiin tarkoitettun aliohjelmakirjaston ja on yhteensopiva kaikkien yleisimpien palvelinohjelmistojen kanssa.

4.6 MySQL-tietokanta ja SQL-kyselykieli

MySQL on avoimeen lähdekoodiin perustuva relaatiotietokantaohjelmisto, jota käytetään laajalti erilaisissa tietojen järjestelmällistä tallentamista vaativissa web- ja muissa sovelluksissa. Relaatiotietokannoissa tiedot tallennetaan taulukkomaisesti; tallennettavaa tietoa varten määritellään taulu jolle nimetään tarpeellinen määrä sarakkeita sekä asetetaan näiden tietotyypit. Tietokannan taulujen väleille voidaan asettaa yhteyksiä, jolloin jokin taulun sarakkeista toimii ns. vierasavaimena, joka voidaan yhdistää vastaavaan tietoon toisessa taulussa.

MySQL tietokantaa ohjataan SQL-kyselykielellä. SQL-kyselyillä voidaan hakea, lisätä, poistaa tai muokata tietoja sekä hallita tietokannan rakennetta ja asetuksia. Kyselyitä voidaan tehdä suoraan MySQL-komentorivityökalulla, tai ohjelmallisesti esimerkiksi PHP-kielen avulla. Tietokantaa käytävissä web-sovelluksissa onkin tarpeellista rakentaa ohjelmallinen rajapinta sovelluksen ja tietokannan välille.

Tässä työssä MySQL-tietokantaa tullaan käyttämään sovelluksen käyttäjä-, harjoitus- ja muiden tietojen tallentamiseen web-palvelimelle.

5 OHJELMISTON SUUNNITTELU

5.1 Käyttöliittymä

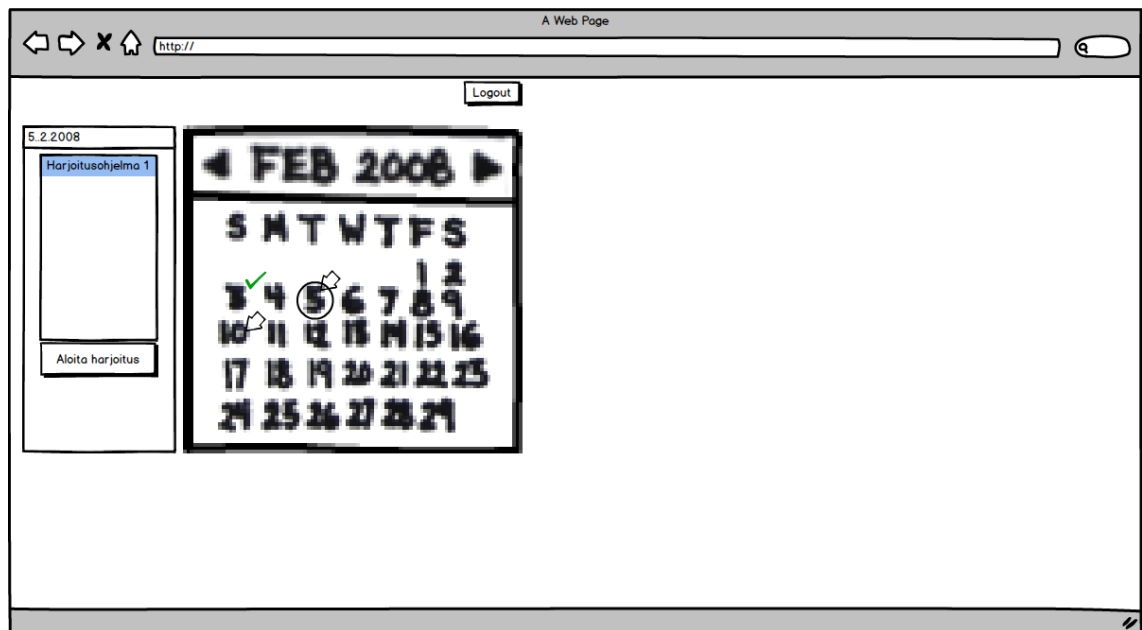
Sovellukseen suunniteltiin erinäisiä käyttöliittymänäkymiä sekä harjoitusohjelmia luovien fysioterapeuttien että niitä suorittavien asiakkaiden käytettäväksi. Käyttöliittymäkuvien luomiseen käytettiin Balsamiq Mockups ohjelmaa.

Koska harjoitusohjelmien suorittaminen pyritään mahdollistamaan mahdollisimman hyvin mobiililaitteilla, on siinä käytettävistä näkymistä suunniteltu näille laitteille soveltuvan kokoiset. Fysioterapeuttien ei kuitenkaan odoteta juuri luovan harjoitusohjelmia mobiililaitteilla, joten kyseisten käyttöliittymien suunnitelmissa on käytetty enemmän tilaa.

5.1.1 Kalenterinäkymä

Kalenterinäkymä on harjoituksia suorittavan käyttäjän päänäkymä, joka avautuu ensimmäisenä käyttäjän kirjaututtua sisään sovellukseen. Näkymässä käyttäjä voi tarkastella hänelle määritettyjä harjoitusohjelmia sekä aloittaa niiden suorittamisen.

Kuvassa 1 näkyvästä kalenterista käyttäjä näkee merkittynä päivät joille hänelle on määritelty harjoitusohjelmia tehtäväksi. Päivän kohdalla oleva kuvake muuttuu, kun käyttäjä on suorittanut kaikki kyseisen päivän harjoitusohjelmat. Vasemmalla olevasta listasta käyttäjä näkee valitsemansa päivän harjoitusohjelmat ja voi aloittaa niiden suorittamisen.

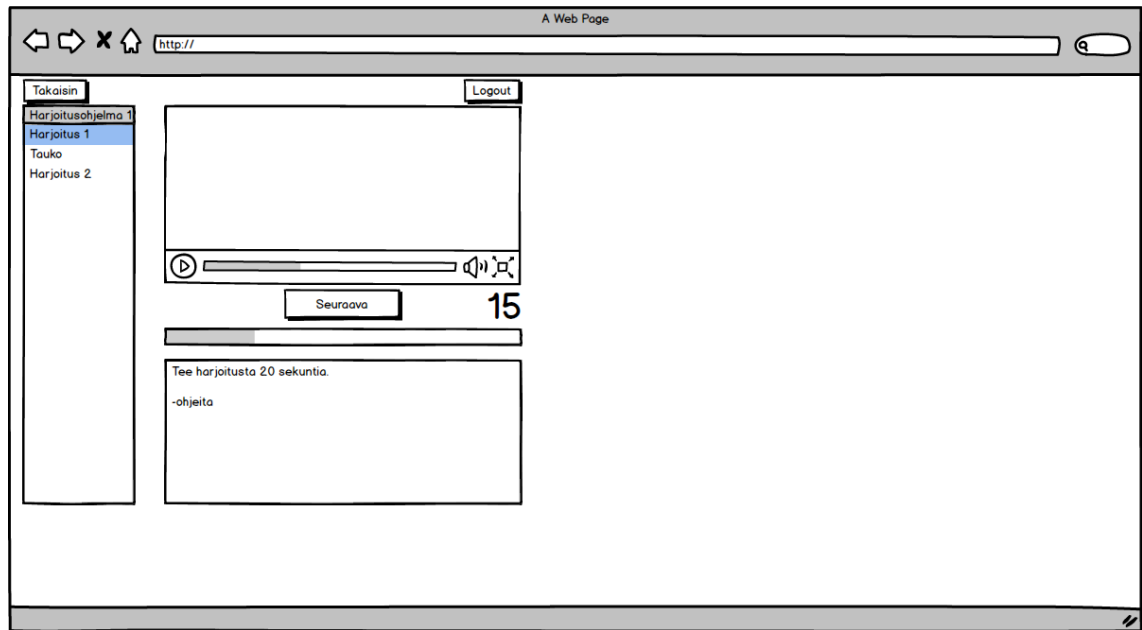


KUVA 1: Kalenterinäkymän käyttöliittymäsuunnitelma

5.1.2 Harjoitusnäkyvä

Harjoitusnäkyvässä (kuva 2) käyttäjä ohjeistetaan suorittamaan aloittamansa harjoitusohjelman harjoitukset. Näkymän tärkeimmät visuaaliset elementit ovat käynnissä olevan harjoituksen ohjevideo ja -teksti. Ajustetuissa harjoituksissa näytetään myös jäljellä oleva aika sekä palkkina että numerona. Sivulla olevassa listassa näytetään koko harjoitusohjelman sisältö.

Koska käyttäjän odotetaan suorittavan fyysisiä harjoituksia tässä näkyvässä ollessaan, on käyttäjältä vaaditun syötteen määrä sovellukselle pyritty minimoimaan. Käyttöliittymän ainoa käyttäjältä harjoitusten aikana toimia vaativa elementti on kuvassa näkyvä 'Seuraava'-painike, jolla käyttäjä voi siirtyä harjoitusohjelmassa eteenpäin. Muita kuvassa näkyviä painikkeita ('Takaisin' ja 'Logout') käytetään vain näkyvästä tai sovelluksesta poistumiseen.

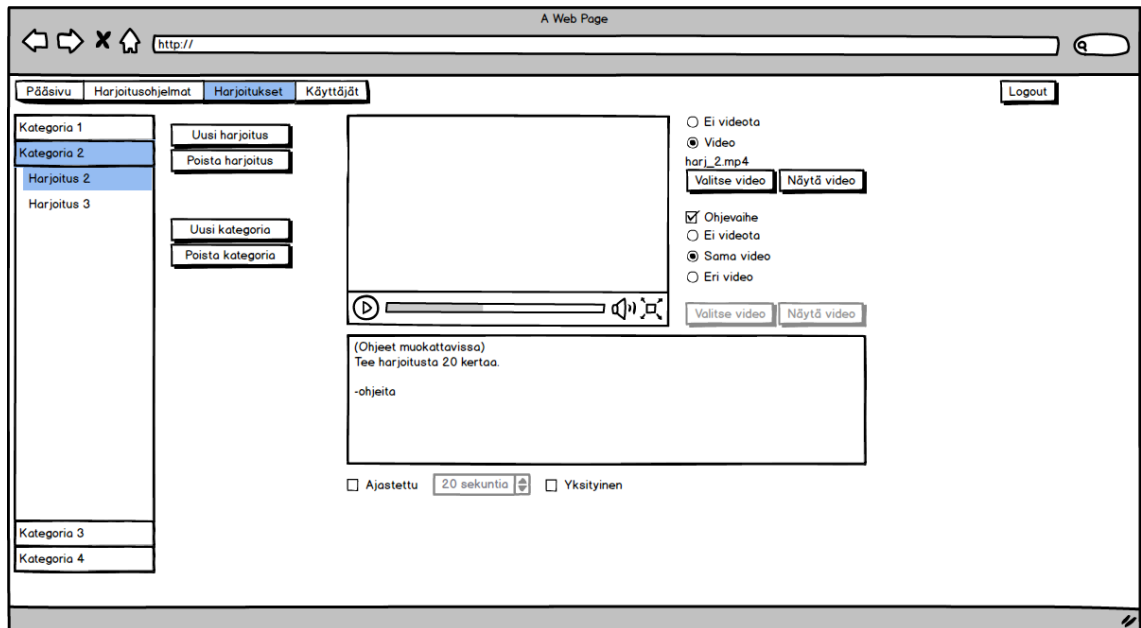


KUVA 2: Harjoitusnäkmän käyttöliittymäsuunnitelma

5.1.3 Harjoitusten ja harjoitusohjelmien luomisenäkymät

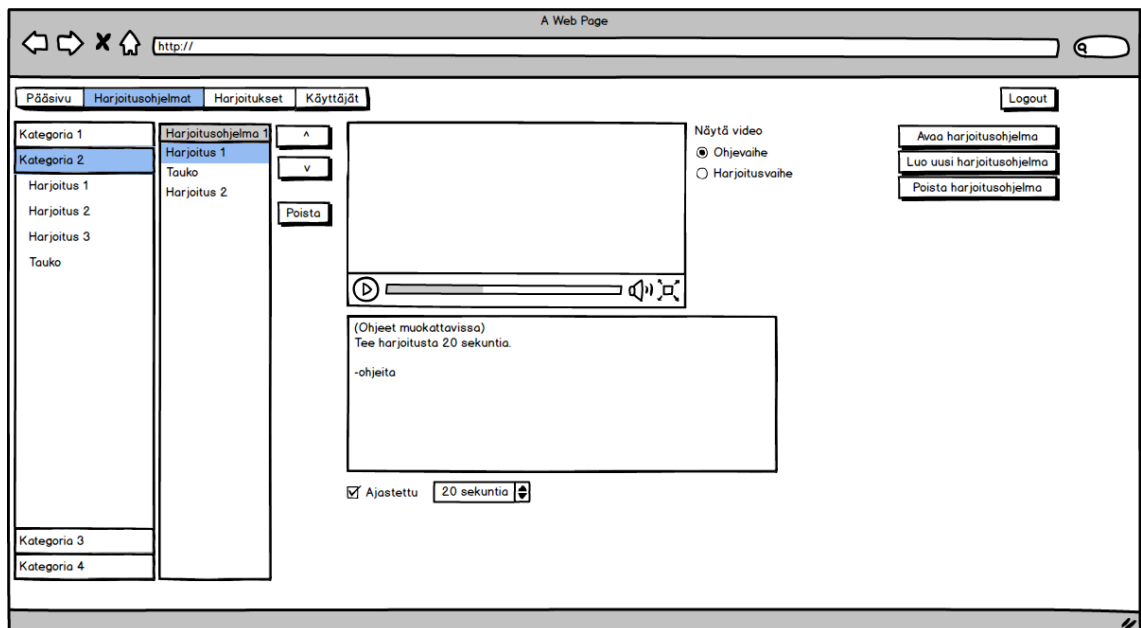
Harjoitusten ja harjoitusohjelmien luomisenäkymissä fysioterapeutti voi luoda yksittäisiä harjoituksia sekä koostaa harjoituksista harjoitusohjelmia.

Kuvassa 3 esitetyssä käyttöliittymässä voidaan luoda ja hallita sovellukseen tallennettuja harjoituksia sekä harjoituskategorioita. Kuvassa vasemmalla olevasta harjoituslistasta valitaan muokattava harjoitus, jolloin sille voidaan asettaa ohjevideot, ohjeteksti ja muut parametrit muilla kuvassa olevilla käyttöliittymäelementeillä.



KUVA 3: Harjoitusten luomisenäkymän käyttöliittymäsuunnitelma

Harjoitusohjelmien luomiseen käytettävässä näkymässä ohjelma koostetaan valitsemalla kuvan 4 vasemmassa laidassa näkyvästä listasta yksittäisiä harjoituksia, jotka lisätään muokattavana olevaan ohjelmaan. Harjoitusohjelman sisältämät harjoitukset nähdään harjoituslistan vieressä olevasta listasta ja niitä on mahdollista poistaa sekä niiden järjestystä muuttaa käyttöliittymässä olevilla painikkeilla. Näkymässä on myös käyttöliittymäelementit harjoitusten ohjevideoiden tarkasteluun sekä ohjetekstien ja ajastusparametrien harjoitusohjelmakohtaiseen muokkaukseen.



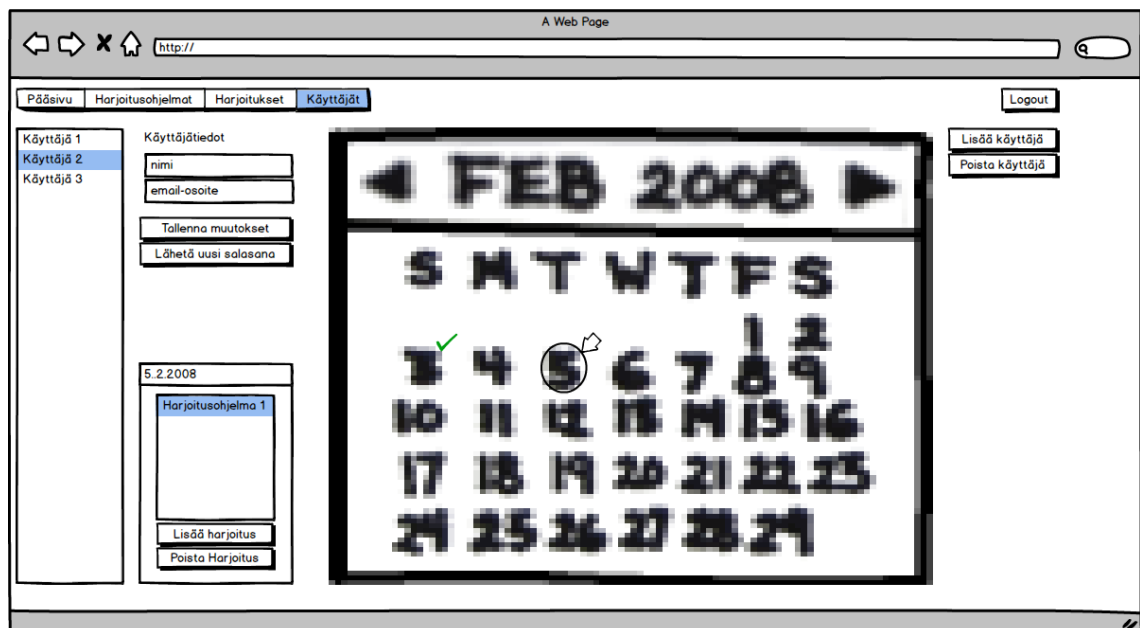
KUVA 4: Harjoitusohjelmien luomisenäkymän käyttöliittymäsuunnitelma

5.1.4 Harjoitusohjelmien ajastus ja käyttäjähallintanäkymä

Käyttäjähallintanäkymässä fysioterapeutti voi luoda asiakkailleen käyttäjätunnuksia, hallinnoida heidän käyttäjätietojaan sekä asettaa heille harjoitusohjelmia suoritettavaksi haluttuina aikoina.

Kuvassa 5 vasemmalla näkyvästä listasta valitaan käyttäjä, jonka tietoja halutaan hallita. Valitun käyttäjän tiedot tulevat näkyviin niille tarkoitettuihin tekstikenttiin, joista niitä on mahdollista muokata.

Kuvassa näkyvään kalenteriin on merkitty päivät, joille valitulle käyttäjälle on ajastettu harjoitusohjelmia. Päivät, joiden kaikki harjoitusohjelmat käyttäjä on suorittanut, on merkitty eri kuvakkeella. Valitsemalla kalenterista haluamansa päivän, fysioterapeutti voi lisätä käyttäjälle harjoitusohjelmia kyseiselle päivälle tai muokata päivälle jo asetettuja ohjelmia.



KUVA 5: Ajastus ja käyttäjähallintanäkymän käyttöliittymäsuunnitelma

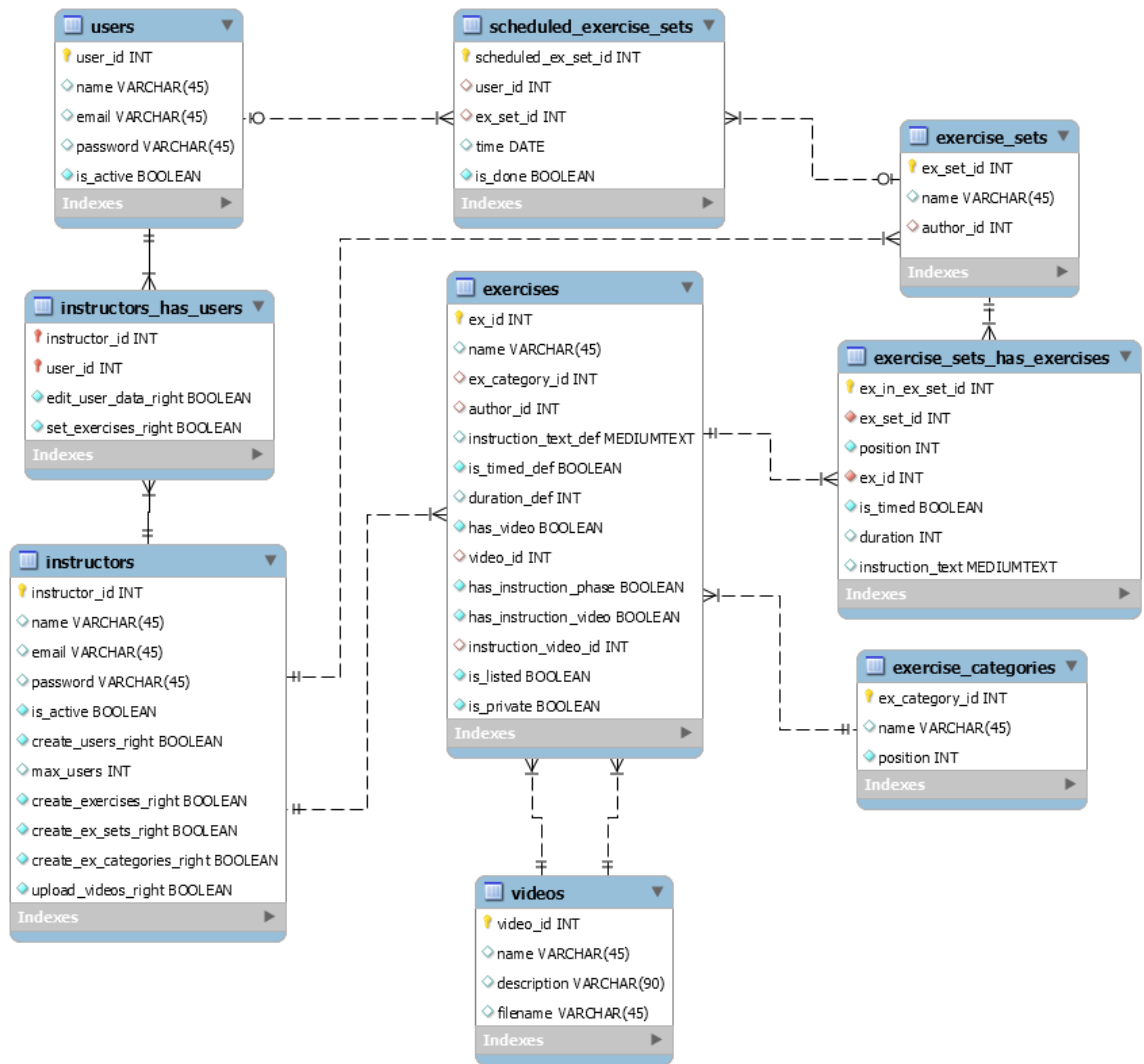
5.2 Tietokanta

Sovellukselle on sen käyttämien tietojen tallennusta varten tarpeellista luoda tietokanta. Tietokantana käytetään aiemmin esiteltyä MySQL- relaatiotietokantaa. Tietokannan suunnittelun apuna käytettiin MySQL Workbench ohjelmaa. Kuvassa 6 on esitetty tietokannan malli.

Koska sovellusta käyttävät kaksi eri käyttäjäryhmää, fysioterapeutit ja heidän asiakkaansa, eivät käytä juurikaan samoja sovelluksen ominaisuuksia, on käytännöllistä luoda käyttäjäryhmille erilliset taulut (kaaviossa taulut 'instructors' ja 'users'). Fysioterapeuteille tallennetaan normaalien käyttäjätietojen lisäksi tiedot käyttöoikeuksista sovelluksen yksittäisiin ominaisuuksiin ja asiakkaiden maksimimäärä. Fysioterapeuttien hallinnoimat asiakkaat ja oikeudet muokata yksittäisten asiakkaiden tietoja tallennetaan liittotauluun ('instructors_has_users'); vaikka tässä työssä yksittäisellä asiakkaalla voi olla vain yksi fysioterapeutti, jolloin liittotaulua vaativaa relaatiota ei tarvita, on myöhemmän kehityksen kannalta hyödyllistä sisällyttää mahdollisuus useampaan.

Harjoitukset ja harjoitusohjelmat tallennetaan omiin tauluihinsa ('exercises' ja 'exercise_sets'), joihin tallennetaan varsinaisen sisällön ja parametrien lisäksi tieto harjoituksen tai ohjelman luoneesta fysioterapeutista. Harjoitusten listauksen avuksi luodaan harjoituskategoriat sisältävä taulu ('exercise_categories'). Myös käytettävissä olevien harjoitusvideoiden tiedot on syytä tallentaa omaan tauluunsa ('videos'). Harjoitusohjelmien sisältämät harjoitukset tallennetaan käyttäen näiden taulujen välistä liittotaulua ('exercise_sets_has_exercises'). Liitostauluun tallennetaan harjoitusten järjestyksen lisäksi harjoituksen harjoitusohjelmakohtaiset parametrit, jotka korvaavat itse harjoituksen oletusparametrit.

Tiedot käyttäjille määritellyistä harjoitusohjelmista, niiden ajankohdista ja tieto ohjelmien suorittamisesta tallennetaan omaan tauluunsa ('scheduled_exercises').



KUVIA 6: Tietokannan ER-kaavio

6 TEKNINEN TOTEUTUS

6.1 Web-käyttöliittymien toteutus

Sovelluksen web-käyttöliittymistä toteutettiin prototyyppiversiot, jotka sisältävät aiemmin esiteltyt ominaisuudet.

6.1.1 Sisäänkirjautumissivu

Sovelluksen sisäänkirjautumissivulle luotiin tekstikentät käyttäjän sähköpostia ja salasanaa varten sekä valintapainikkeet käyttäjätyyppin (fysioterapeutti tai asiakas) valintaan. Sisäänkirjautuessa sivun JavaScript-koodi (ohjelmakoodi 1) lähettää tiedot käyttäjätyyppin mukaiselle PHP-tiedostolle, joka tarkistaa tiedot tietokannasta ja palauttaa tarkistuksen tuloksen sivulle. Tuloksen perusteella kirjautumissivu joko siirtää käyttäjän oikeaan näkymään tai ilmoittaa sisäänkirjautumissivulle virheellisistä tiedoista.

```
$.post(($("#userRadio").prop("checked")) ? "php/user_login.php" :
"php/instructor_login.php",
{
email: $("#inputEmail").val(),
password: $("#inputPassword").val()
},
function(data, status){
if(data=="1"){
window.location.assign(($("#userRadio").prop("checked")) ?
"fysio_user.php" : "fysio.php");
}
else{
alert("Väärä sähköposti tai salasana");
}
});
```

OHJELMAKOODI 1: Sisäänkirjautumistietojen lähetys ja käsittely

Sisäänkirjautumisen yhteydessä PHP-koodi (ohjelmakoodi 2) asettaa käyttäjän tunnistenumeron ja sisäänkirjautumistiedon erityisiin istuntomuuttujiin, joita sovelluksen muut osat voivat hyödyntää kunnes käyttäjä sulkee web-selaimen tai kirjautuu ulos. Kaikki muut näkymät sovelluksessa on ohjelmoitu siirtämään käyttäjä kirjautumissivulle, mikäli näitä tietoja ei ole asetettu.

```

if($result){
    $output = mysqli_fetch_object($result);

    session_start();
    $_SESSION['login']="1";
    $_SESSION['id']=$output->instructor_id;
    echo 1;
}
else{
    echo 0;
}

```

OHJELMAKOODI 2: Istuntomuuttujien asetus

6.1.2 Siirtyminen fysioterapeutin käyttöliittymänäkymien välillä

Fysioterapeutin käyttöliittymässä siirtyminen eri näkymien välillä tapahtuu käyttöliittymän ylälaudassa olevan navigointipalkin avulla. Palkin painikkeet on asetettu sivun JavaScript-koodissa (ohjelmakoodi 3) lataamaan haluttu näkymä sekä merkitsemään näkymän painike aktiiviseksi jQuery-funktioiden avulla.

```

$("#usersButton").click(function() {
    $("#mainContainer").load("users.php");
    $(currentNavId).attr("class", "inactive");
    $("#usersNav").attr("class", "active");
    currentNavId="#usersNav";
});

```

OHJELMAKOODI 3: Esimerkki siirtymistoimintojen asettamisesta painikkeeseen

Käyttöliittymänäkymiä ei avata selaimen uusina sivuina, vaan ladataan sovelluksen fysioterapeuttien käyttämän käyttöliittymän pohjana olevalle sivulle asetettuun elementtiin. Koska kaikki näkymät ladataan samalle sivulle, voidaan tämän sivun JavaScript-muuttujiin tallennettua tietoa helposti käyttää kaikissa näkymissä, eikä erillistä tiedon siirtämistä näkymien välillä tarvita.

6.1.3 Harjoitusten ja harjoitusohjelmien luomisnäkyvien toiminnot

Harjoitusohjelmien luomisnäkyvässä (kuva 7) ladataan aluksi käytettävissä olevat harjoituskategoriat tietokannasta. Jokaiselle kategorialle luodaan dynaamisesti HTML-elementtien koodit, jotka asetetaan jQuery-funktion avulla listalle varattuun elementtiin

(ohjelmakoodi 4). Tämän jälkeen kategorioidiin haetaan niihin kuuluvat harjoitukset tietokannasta, jotka luodaan HTML-elementeiksi ja asetetaan aiemmin luotuihin kategorioiden elementteihin samalla tavalla. Harjoitusten HTML-elementit toimivat painikkeina, jotka lisäävät kyseisen harjoituksen avoimena olevaan harjoitusohjelmaan.

```

var content = "";

$.each(categories, function(i, category){
    content += '<div class="panel panel-default">' +
    '<div class="panel-heading">' +
    '<h4 class="panel-title">' +
    '<a data-toggle="collapse" data-parent="#exList"' +
    'href="#cat'+
    category.ex_category_id +'>' + category.name +'</a></h4>' +
    '</div>' +
    '<div id="cat'+ category.ex_category_id +
    '" class="panel-collapse collapse">' +
    '</div>' +
    '</div>';
});
$("#exList").html(content);

```

OHJELMAKOODI 4: Katogoriaelementtien dynaaminen luominen

Muokattavissa olevat harjoitusohjelmat haetaan tietokannasta käyttöliittymässä olevaan pudotusvalikkoon, josta valitsemalla ne avataan. Avatessa ohjelman sisältämät harjoitukset haetaan tietokannasta ja asetetaan ohjelmaa kuvaavaan listaan, josta voidaan avata yksittäisiä harjoituksia muokattavaksi. Ohjelman nimi asetetaan muokattavaan tekstikenttään, joka on asetettu automaattisesti päivittämään nimen tietokantaan käyttäjän muuttaessa sitä (ohjelmakoodi 5).

```

$("#setNameInput").change(function(){
    if(currentExSetId != -1){
        saveSetName(currentExSetId, $(this).val());
    }
});

```

OHJELMAKOODI 5: Automaattisen tallennuksen asettaminen tekstikenttään

Avatessa yksittäisiä harjoituksia, muokattavissa olevat tiedot haetaan niille tarkoitettuihin elementteihin ja harjoitukseen liittyvät videot ladataan katsottavaksi. Sekä ohjetekstiä että ajastusasetuksia muutettaessa muutokset tallennetaan automaattisesti tietokantaan samalla tavoin kuin aiemmin mainittu harjoitusohjelman nimi. Avoimena olevan harjoituksen ohjelmasta poistava ja sijaintia harjoitusohjelmassa muuttavat painikkeet asetetaan harjoitusta avattaessa suorittamaan tarvittavat muutokset tietokantaan (ohjel-

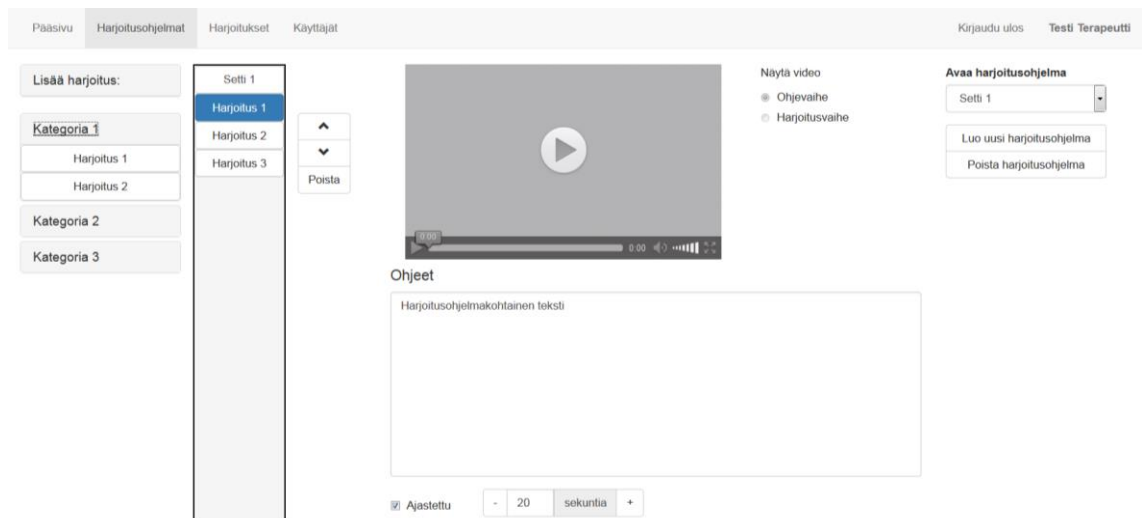
makoodi 6). Näiden muutosten jälkeen harjoitusohjelma ladataan uudelleen, jotta käyttöliittymässä näkyvä ohjelman rakenne vastaisi tietokannassa olevaa.

```

$("#deleteExeButton").attr("onClick",
"deleteExeFromSet (" + exInSetId + ")");
$("#moveBackButton").attr("onClick",
(exe.position !== maxExPos) ? "moveExe (" + exInSetId + ", "+ 1 + ") " : "");
$("#moveFwdButton").attr("onClick",
(exe.position !== 1) ? "moveExe (" + exInSetId + ", "+ -1 + ") " : "");

```

OHJELMAKOODI 6: Toimintojen asetus ohjelman rakennetta muuttaviin painikkeisiin



KUVA 7: Harjoitusohjelmien luomisenäkymä

Yksittäisten harjoitusten luomisenäkymän toiminnot ovat pääosin samankaltaisia kuin harjoitusohjelmien luomisenäkymässä ja ne on toteutettu käyttäen samaa koodia joillain muutoksilla. Ainoat huomattavan erilaiset ominaisuudet ovat harjoitusten ohjevideoiden valinta ja uusien videoiden lataaminen, joita ei kuitenkaan ole tätä raporttia kirjoitettaessa saatu valmiiksi.

6.1.4 Käyttäjähallinnan ja käyttäjän kalenterinäkömän toiminnot

Käyttäjähallintanäkymässä (kuva 8) käyttäjälistan koostaminen sekä käyttäjien avaaminen, lisääminen ja poistaminen tapahtuvat samoilla periaatteilla kuin harjoitusohjelmien vastaavat aiemmin esitellyt toiminnot. Käyttäjätietojen muokkaus tapahtuu myös samoin kuin harjoitustietojen, lukuun ottamatta tietojen tallennusta tietokantaan joka ei tapahdu automaattisesti.

Käyttäjänhallinnan kalenterielementin toteuttamiseen on käytetty Bootstrap-käyttöliittymiin tarkoitettua, ilmaista Zabuto Calendar- lisäosaa (Heijnen 2013). Lisäosaa on muokattu suomenkielisen kalenterin lisäämiseksi. Käyttäjän tietoja avatessa kalenteriin merkitään kaikki päivät joille käyttäjälle on asetettu harjoitusohjelmia (ohjelmakoodi 7). Koska samalla päivällä voi olla useampia harjoitusohjelmia ja päivälle asetettava merkki määräytyy viimeisen käsitellyn ohjelman mukaan, järjestetään suoritettut ohjelmat tietokannasta haettaessa ensimmäisiksi. Tällöin viimeinen käsiteltävä ohjelma voi olla suoritettu vain, jos kaikki päivän ohjelmat ovat suoritettuja.

```
$.post(queryUrl, {
  request: "getScheduledSets",
  userId: userId
},
function(data2, status2){
  var sets = jQuery.parseJSON(data2);
  var setData = [];
  $.each(sets, function(i, set){
    setData[i] = {
      "date": set.time,
      "badge": (set.is_done) ? "success" : true,
      "footer": set.is_done
    };
  });
  loadCalendar(setData);
});
```

OHJELMAKOODI 7: Harjoitusohjelmien merkitseminen kalenteriin

Päivän valitseminen kalenterista on asetettu suorittamaan funktio, joka hakee kyseisen päivän harjoitusohjelmat ja luo niistä dynaamisesti listan, jossa suoritettut ohjelmat on merkitty merkillä. Listaan harjoitusohjelmia lisäävät ja poistavat painikkeet ja valitsin on toteutettu aiemmin esitellyillä tavoilla.

Uuden salasanan luominen ja lähettäminen käyttäjälle hoidetaan PHP-tiedostossa, jolle näkymä lähettää tiedon painiketta painettaessa. PHP-koodi luo satunnaisen salasanan, asettaa käyttäjän aktiiviseksi ja lähettää tiedon käyttäjän sähköpostiosoitteeseen (ohjelmakoodi 8).

```
$password = substr(str_shuffle(MD5(microtime())), 0, 8);
```

```
$stmt = mysqli_prepare($con, "UPDATE users
SET password = ?, is_active = 1
WHERE user_id = ?");
mysqli_stmt_bind_param($stmt, 'si', $password, $_POST['userId']);
mysqli_stmt_execute($stmt);
```

```
mail($_POST['email'], "Tunnus luotu", "Sinulle on luotu käyttäjätunnus
palveluun. Salasana: ".$password);
```

OHJELMAKOODI 8: Uuden salasanan luonti ja lähetys käyttäjälle

The screenshot displays a user management interface. On the left, there is a sidebar with 'Käyttäjät' (Users) and 'Uusi käyttäjä' (New user). The main content area shows the profile for 'Testi Käyttäjä' with fields for 'Nimi' (Name) and 'Sähköpostiosoite' (Email address). Below these are buttons for 'Tallenna muutokset' (Save changes) and 'Laheta uusi salasana' (Send new password). A calendar for '20.4.2015' is shown, with a dropdown menu for 'Setti 1' and buttons for 'Lisää harjoitusohjelma' (Add exercise program) and 'Poista harjoitusohjelma' (Remove exercise program). A calendar grid for 'Huhtikuu 2015' is also visible, with the 21st highlighted.

KUVA 8: Käyttäjähallintanäkymä

Käyttäjän kalenterinäkymä käyttää samoin toteutettua kalenterielementtiä kuin käyttäjähallintanäkymä. Suoritettavan harjoitusohjelman valitseminen valitun päivän ohjelmat näyttävästä listasta siirtää käyttäjän harjoitusten esitysnäkymän sisältävälle sivulle ja lähettää harjoitusohjelman tunnustenumeron osana sivun osoitetta.

6.1.5 Harjoitusten esitysnäkymän toiminnot

Harjoitusten esitysnäkymässä ladataan aluksi harjoitukset sivulle lähetetyn harjoitusohjelman tunnustenumeron perusteella. Harjoitusten nimet asetetaan listaan ja tunnustenumeroista luodaan taulukkomuuttuja, jonka perusteella yksittäiset harjoitukset voidaan ladata.

Yksittäisiä harjoituksia avattaessa varsinaisen harjoitusvaiheen tiedot tallennetaan olioon ja harjoituksen ohjeteksti sekä mahdollinen ohjevideo asetetaan suoraan niille tarkoitettuihin elementteihin. Harjoitusvaiheen alkaessa, joko käyttäjän toimesta tai, mikäli harjoitukseen ei ole määritelty ohjevaihetta, automaattisesti, haetaan harjoituksen tiedot aiemmin mainitusta oliosta. Jos harjoitus on ajastettu, asetetaan ajastinpalkki mittamaan harjoitukselle määriteltyä aikaa ja sovellus siirtymään automaattisesti seuraavaan harjoitukseen ajan loppuessa (ohjelmakoodi 9).

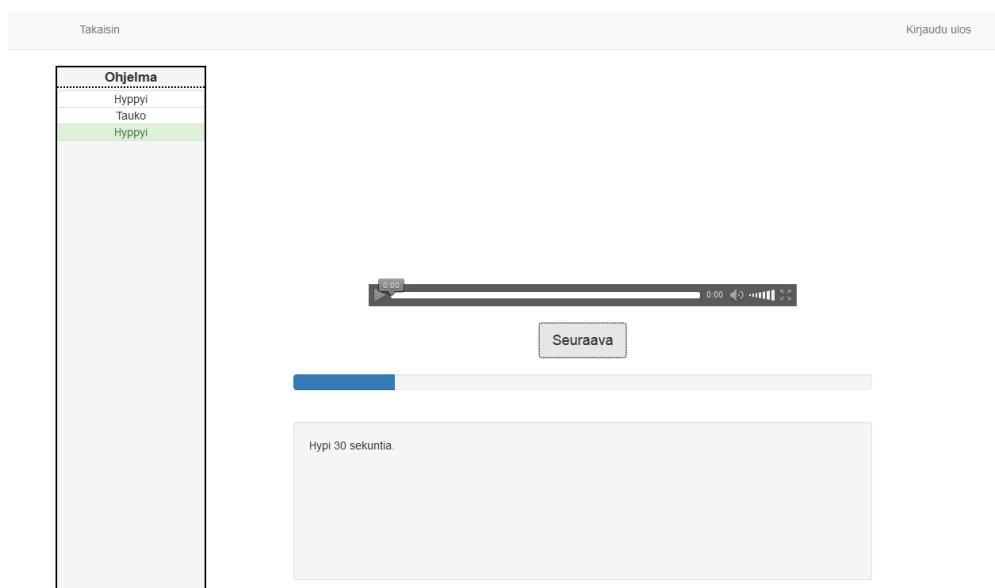
```

if(exPhase.isTimed) {
    $("#progBar").prop("hidden", false);
    $("#timeBar").animate({width: "100%"},
        exPhase.duration*1000, function() {
            nextExe();
        });
}

```

OHJELMAKOODI 9: Ajastinpalkin toimintojen asettaminen

Kun kaikki harjoitukset harjoitusohjelmassa on suoritettu, ohjaa sovellus käyttäjän automaattisesti takaisin kalenterinäkymään ja merkitsee harjoitusohjelman suoritetuksi tietokantaan.



KUVA 9: Harjoitusten esitys näkymä

6.2 Tietokantayhteyksien toteutus

Sovelluksen tietokantayhteydet on pääosin toteutettu yksittäiseen PHP-tiedostoon, jolle web-käyttöliittymä lähettää pyynnön ja tarvittavat parametrit. Tiedosto suorittaa tarvittavat kyselyt tietokantaan ja palauttaa tarvittaessa tulokset JSON-muodossa käyttöliittymälle.

PHP-tiedostoon on listattu kaikki sovelluksessa tarvittavat kyselyt, joista valitaan pyydetty kysely käyttöliittymän lähettämän 'request'-parametrin perusteella. Kyselystä valmistellaan SQL-lause johon asetetaan käyttöliittymän lähettämät parametrit (ohjelmakoodi 10). Mikäli kysely vaatii tiedon hakemista tietokannasta ennen varsinaisen pyydetyn kyselyn suoritusta, suoritetaan tarvittavat kyselyt ensin ja tarvittavat tiedot asetetaan muuttujiin.

```
switch($_POST['request']){
case 'getInstructor':
    $stmt = mysqli_prepare($con, "SELECT * FROM instructors
    WHERE instructor_id = ?");
    mysqli_stmt_bind_param($stmt, 'i', $_POST['instructorId']);
break;
```

OHJELMAKOODI 10: Esimerkki SQL-lauseen valinnasta ja valmistelusta

SQL-lauseen valmistelun jälkeen lause suoritetaan ja, mikäli kysely palauttaa tietoja, tuloksista muodostetaan taulukkomuuttuja, joka muunnetaan JSON-muotoon. Tiedot palautetaan tämän jälkeen käyttöliittymälle (ohjelmakoodi 11).

```
mysqli_stmt_execute($stmt);

$result = mysqli_stmt_get_result($stmt);
if(mysqli_num_rows($result) == 0){
    $output = "";
}

while($row = mysqli_fetch_array($result)){
    $output[] = $row;
}

echo(json_encode($output, JSON_FORCE_OBJECT));
mysqli_close($con);
```

OHJELMAKOODI 11: SQL-lauseen suoritus ja tietojen palautus

7 YHTEENVETO

Työtä aloittaessani tietoni ja kokemukseni web-ohjelmoinnista oli melko vähäistä. Sovelluksesta suunniteltiin aluksi toteutettavaksi mobiilialustalle, mutta web-ratkaisu todettiin lopulta tarkoitukseen sopivammaksi. Tästä johtuen siirtyminen sovelluksen suunnitteluvaiheesta toteutusvaiheeseen oli melko haasteellista ja sopivien tekniikoiden valinta vaati huomattavan määrän tiedonhankintaa. Tästä huolimatta itse toteutus sujui pääosin hyvin; opin tarvittavat tekniikat nopeasti eikä ylitsepääsemättömiä ongelmia ilmennyt.

Kokonaisen ohjelmistoprojektin suunnittelu ja toteuttaminen itsenäisesti oli uusi ja varsin opettavainen kokemus. Erityisesti perussuunnittelun merkitys korostui itselleni uudella tavalla. Suunnittelu eteni aluksi hyvin hitaasti, mutta sovelluksen perusrakenteen hahmotuttua projektin eteneminen saavutti eräänlaisen kriittisen massan jonka jälkeen edistystä alkoi tapahtua huomattavasti nopeammin. Koenkin, että tämä opinnäytetyö kehitti huomattavasti kykyäni laajempien sovelluskehitysprosessien hahmottamiseen ja käytännön toteuttamiseen.

Työssä luodun prototyypisovelluksen toteutus onnistui mielestäni hyvin. Sovellukseen saatiin toteutettua suurin osa tärkeimmistä suunnitelluista ominaisuuksista, eikä mitään suurempia vikoja niiden toiminnassa ole ilmennyt. Myös käyttöliittymälle tavoiteltu selkeys ja helppokäyttöisyys saavutettiin. Useita pieniä, mutta sovelluksen käytössä tarpeellisia, ominaisuuksia jouduttiin kuitenkin jättämään toteuttamatta, jotta käytettävissä oleva aika ja resurssit riittäisivät tärkeimpien ominaisuuksien toteuttamiseen.

LÄHTEET

Heijnen, A. 2013. Zabuto Calendar README-tiedosto.

Mozilla Developer Network. 2015a. Introduction to HTML. Luettu 25.3.2015.
<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction>

Mozilla Developer Network. 2015b. JavaScript Guide. Luettu 25.3.2015.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>

Murphey, R. 2012. jQuery Fundamentals. Luettu 25.3.2015.
<http://jqfundamentals.com/legacy/>

Olson, P. (toim.) 2015. PHP manual. PHP Documentation Group. Luettu 25.3.2015.
<http://php.net/manual/en/>

PhysioTools. 2015. Luettu 3.2.2015.
<http://physiotools.com>

Päivittäiset Treenit. 2015. Luettu 3.2.2015.
<https://play.google.com/store/apps/details?id=com.tinymission.dailyworkoutsfree>

Training Apps. 2015. Luettu 3.2.2015.
<https://play.google.com/store/apps/details?id=fi.trainingapps.app>

W3C. 2014. HTML5 is a W3C recommendation. Luettu 25.3.2015.
<http://www.w3.org/blog/news/archives/4167>

w3schools.com. 2015. Bootstrap Get Started. Luettu 25.3.2015.
http://www.w3schools.com/bootstrap/bootstrap_get_started.asp

WebPlatform.org. 2015. What is CSS? Luettu 25.3.2015.
https://docs.webplatform.org/wiki/tutorials/learning_what_css_is

Workout Trainer. 2015. Luettu 3.2.2015.
<https://play.google.com/store/apps/details?id=com.skimble.workouts>