

Jani Kärkkäinen

REFERENSSILUETTELOTIETOKANTA

REFERENSSILUETTELOTIETOKANTA

Jani Kärkkäinen
Opinnäytetyö
Syksy 2019
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä: Jani Kärkkäinen

Opinnäytetyön nimi: Referenssiluettelotietokanta

Työn ohjaaja: Jukka Jauhiainen

Työn valmistumislukukausi ja -vuosi: Syksy 2019

Sivumäärä: 28

Työn tarkoituksena oli korvata Linja Arkkitehdit-yrityksen projektireferensseihin käyttämä Excel-
taulukko tietokantapohjaisella ratkaisulla. Työssä tuli suunnitella ja toteuttaa tietokanta ja sille käyt-
töliittymä. Referenssitietoja käytetään yrityksen tarjoustoiminnassa, ja käyttöliittymässä tuli olla
suodattimia ja toimintoja helpottamaan sitä.

Käyttöliittymä toteutettiin selainpohjaisena. Sen ohjelmoinnissa käytettiin HTML-, JavaScript-,
PHP-, SQL- ja CSS-kieliä sekä Bootstrap- ja Bootstrap Table-kirjastoja. Tietokantana käytettiin
MySQL-tietokantaa, joka toimii yrityksen sisäisessä verkossa heidän omalla palvelimellaan.

Työ tehtiin keväällä 2019 ja referenssitietokanta tullaan ottamaan käyttöön lähitulevaisuudessa.

Asiasanat: tietokannat, käyttöliittymät, ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software development

Author: Jani Kärkkäinen

Title of thesis: Reference list database

Supervisor: Jukka Jauhiainen

Term and year when the thesis was submitted: Fall 2019

Number of pages:28

Subject of the thesis was to replace an Excel document with database-based solution. The Excel document has been used by a company called Linja Arkkitehdit for their project references. The goal was to design and create a database and user interface for it. The company uses reference data to make offers for customers and the user interface should have features and filters to support that.

The user interface was developed as a web-based interface. Programming languages such as HTML, JavaScript, PHP, SQL and CSS were used to create it. Component libraries such as Bootstrap and Bootstrap Table were also used. MySQL database was used as database in this project and it is located in the company's own server environment.

Thesis was done in 2019 Spring and it will be taken to use in near future.

Keywords: database, user interface, programming

SISÄLLYS

| | | |
|-------|-----------------------------------|----|
| 1 | JOHDANTO | 6 |
| 2 | TIETOKANTA | 7 |
| 2.1 | MySQL | 7 |
| 2.2 | Relaatiomalli..... | 7 |
| 3 | KÄYTETYT OHJELMOINTIKIELET | 9 |
| 3.1 | HTML-merkkauskieli..... | 9 |
| 3.2 | JavaScript-ohjelmointikieli | 10 |
| 3.3 | PHP-ohjelmointikieli | 10 |
| 3.4 | SQL-kyselykieli..... | 11 |
| 3.5 | CSS..... | 11 |
| 3.6 | Kehykset ja lisätyökalut | 13 |
| 3.6.1 | Bootstrap..... | 13 |
| 3.6.2 | Bootstrap Table..... | 13 |
| 4 | TOTEUTUS | 14 |
| 4.1 | Käyttöliittymä | 14 |
| 4.2 | Tietokanta..... | 15 |
| 5 | TULOKSET | 17 |
| 5.1 | Tietokannan rakenne..... | 17 |
| 5.2 | Käyttöliittymä | 18 |
| 6 | POHDINTA | 26 |
| | LÄHTEET..... | 27 |

1 JOHDANTO

Linja Arkkitehdit Oy on vuonna 2006 Oulussa perustettu arkkitehtitoimisto. Se on yksi Suomen suurimmista arkkitehtitoimistoista ja sillä on kolme toimipistettä, jotka sijaitsevat Oulussa, Helsingissä ja Jyväskylässä. Yritys tarjoaa monipuolisia, tilaajan tarpeisiin sopeutuvia arkkitehti- ja pääsuunnittelupalveluja. Heidän toimialaansa kuuluvat rakennussuunnittelu- uudis- ja korjaushankkeet kaikkine vaiheineen sekä maankäytön suunnittelu ja kaavoitus. (Linja Arkkitehdit 2019.)

Yritys on aikaisemmin pitänyt referenssiluettelo Excel-taulukossa, joka sisältää perustietoja aikaisemmin tehdyistä projekteista. Projektien määrä on kasvanut niin paljon, että tarjousten ja henkilöesitteiden tekeminen sen pohjalta on tullut työlääksi. Tarjoustoiminnassa täytyy tehdä laatupisteitys, missä otetaan huomioon vain tarjouksen mukaan tietyn vuosimäärän takaiset samankaltaiset projektit, joita yritys on aikaisemmin tehnyt. Henkilöesitteitä käytetään myös tarjoustoiminnassa, kun ehdotetaan tiettyä henkilöä toimimaan jossain roolissa projektissa.

Tavoitteena oli luoda tietokanta ja sille käyttöliittymä, jolla on helpompi tehdä tarvittavia hakutoimintoja ja tulosteita. Ideana oli myös, että jokainen työntekijä voi rajatuilla oikeuksilla päivittää omia tietojaan ja projekteihin osallistumisiaan tietokantaan ja näin ylläpitää itse omaa CV:tä.

2 TIETOKANTA

2.1 MySQL

MySQL on tietokantahallintajärjestelmä, joka käsittelee relaatiotietokantoja. Se on alun perin kehitetty konsultointiyhtiön MySQL AB:n sisäiseen käyttöön. Se on ollut markkinoilla jo yli parikymmentä vuotta ja sitä voidaan käyttää ilmaiseksi GPL-lisenssillä. MySQL sisältää WWW-palveluiden kannalta olennaisia ominaisuuksia ja sillä on jo pitkään toteutettu muun muassa julkaisujärjestelmiä, lomakepalveluita, sähköisen kaupan järjestelmiä ja yritysten internet-palveluja. Se on erityisen suorituskykyinen ja se tukee useita eri ohjelmistokieliä kuten, C, C++, PHP, Perl, Python ja Tcl. Se on helppo asentaa, vaatii vähän ylläpitoa ja on edullinen, mikä tekee siitä hyvän ratkaisun myös pienempien WWW-palvelujen tietokannaksi. (Heinisuo & Rauta 2007, 37–40.)

MySQL kuuluu nykyään Oraclen tuotevalikoimaan ja on kasvanut maailman suosituimmaksi avoimen lähdekoodin tietokantajärjestelmäksi. Sitä käyttävät suuret palvelut kuten, Facebook, Twitter ja YouTube. (Oracle 2019.)

2.2 Relaatiomalli

Relaatiotietokannat perustuvat IBM:n tutkijan E. F. Coddin vuonna 1970 julkaisemaan relaatiomalliin, joka määrittelee relaatiotietokannan teoreettisen pohjan. Relaatiomalli perustuu joukko-oppiin, matematiikkaan ja predikaattilogiikkaan. Relaatiomalli voidaan jakaa kolmeen osaan: rakenne, käsittely ja eheytyssäännöt. (Hovi, Huotari & Lahdenmäki 2005, 7–8.)

Tietokannassa asiakokonaisuudet koostuvat tauluista. Taulut sisältävät sarakkeita ja rivejä. Sarakkeilla on taulun sisällä toisistaan poikkeavat nimet. Sarakkeiden tietojen arvot kuuluvat samaan arvojoukkoon, mikä tarkoittaa, että niille on määritelty yhteinen tietotyyppi ja varattu pituus. Jokaisessa taulussa on perusavain, jolla jokainen rivi voidaan yksilöidä. Tämä tarkoittaa, että perusavaimeksi määritetyllä sarakkeella ei saa olla kahta samaa arvoa. Perusavain voi myös koostua useammasta kuin yhdestä sarakkeesta. Taulujen tietoja voidaan liittää toisiinsa isä-lapsiyhteyden avulla. Isällä voi olla monta lasta, mutta lapsella voi olla vain yksi isä. Tällöin lapsitauluun tehdään linkkikenttä eli viiteavain, joka viittaa isätaulun perusavaimeen. Relaatiomallissa tietoja käsitellään

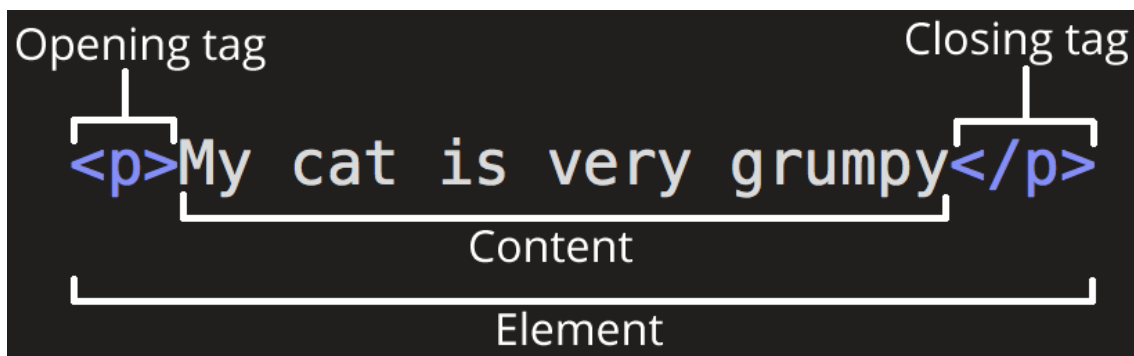
joukko-opillisesti. Taulut koostuvat joukosta rivejä, ja tähän joukkoon voidaan kohdistaa voimakkaita joukko-operaatioita. Nämä operaatiot toteutetaan SQL-kielellä. Operaatioilla voidaan suodattaa taulujen sisältöä tai vaihtaa niiden arvoja tehokkaasti. (Hovi, Huotari & Lahdenmäki 2005, 8–11.)

Relaatiomalli ottaa kantaa myös tietokannan eheyteen. Tietokanta on eheä, kun sen tiedot ovat oikein, ristiriidattomia ja vastaavat reaali maailmaa. Relatiomalliin on määritelty tietyt eheytsrajoitteita. Ensimmäinen niistä on avainehyys sääntö. Sen mukaan taulukon perusavaimen arvo ei saa ikinä olla tyhjä eli NULL. Toinen sääntö on viite-ehyys. Sen mukaan isätaulusta ei saa poistaa rivejä, jotka ovat viitattuina lapsitauluihin, koska tällöin lapsitauluun jäisi niin sanottuja orpoja rivejä, jotka viittaavat isätaulun riveihin, joita ei ole enää olemassa. (Hovi, Huotari & Lahdenmäki 2005, 11–12.)

3 KÄYTETYT OHJELMOINTIKIELET

3.1 HTML-merkkäuskieli

Lyhenne HTML tulee sanoista Hypertext Markup Language. Se ei ole ohjelmointikieli vaan merkkäuskieli, joka määrittelee sisällön rakenteen. HTML-dokumentti koostuu joukosta elementtejä. Elementti sisältää alkutagin, sisällön ja lopputagin (kuva 1). (Korpela & Linjama 2005, 70–72.)

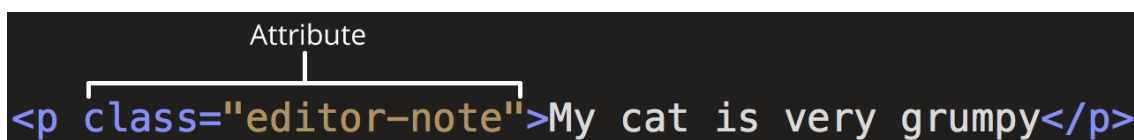


KUVA 1. Esimerkki HTML-kielen kappale-elementistä (<p>). (MDN web docs 2019c.)

Jotkin elementit voivat sisältää myös toisia elementtejä esimerkiksi korostamaan sisältöä, kuten seuraavassa koodi (MDN web docs 2019c).

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

Elementit voivat sisältää myös yhden tai useamman määritteen eli attribuutin (kuva 2). Attribuutit sisältävät lisäinformaatiota elementistä. Attribuutti kirjoitetaan alkutagin sisälle ja se sisältää attribuutin nimen, yhtäläisyysmerkin ja attribuutin arvon, joka on heittomerkkien sisällä. Attribuutit erotetaan toisistaan välilyönnillä. (Korpela & Linjama 2005, 74.)



KUVA 2. Esimerkki HTML-kielen luokka-attribuutista (class). (MDN web docs 2019c.)

Yleisempiä attribuutteja ovat luokat (class). Luokat eivät ole ennalta määriteltyjä vaan sivuntekijä voi itse määrittellä ne. Niiden avulla elementtejä voidaan ryhmittää, mikä on hyödyllistä esimerkiksi, kun määrittellään sivun tyylisääntöjä. Elementti voi kuulua useampaan kuin yhteen luokkaan. Tällöin luokat erotellaan välilyönneillä. Toinen yleinen attribuutti on tunniste(id). Sen avulla elementille voidaan antaa arvo, jolla se pystytään yksilöimään. Tämä on kätevää esimerkiksi, kun halutaan vaikuttaa JavaScriptillä tietyn elementin sisältöön. (Korpela & Linjama 2005, 75–76.)

3.2 JavaScript-ohjelmointikieli

JavaScript on asiakaspuolen (front-end) ohjelmointikieli. Tämä tarkoittaa, että sillä kirjoitetut skriptit on tarkoitettu ajettavaksi käyttäjän selaimessa eikä palvelimella. JavaScript-koodia voi kirjoittaa suoraan HTML-dokumentin sisään <script>-tagien sisään tai omaan tiedostoon. (Negrino & Smith 2007, 2.)

JavaScript on ohjelmointikieli, jota voidaan käyttää lisäämään verkkosivujen interaktiivisuutta. Sillä voidaan tehdä elementtejä, joiden sisältö muuttuu suoraan käyttäjän valintojen mukaan. Sillä voidaan myös muuttaa elementtien ulkonäköä, kun käyttäjä esimerkiksi siirtää hiiren kursorin niiden päälle. Sen avulla voidaan myös vähentää palvelinliikennettä tarkistamalla, ovatko lomakkeeseen syötetyt tiedot oikein jo asiakaspuolen koodissa. JavaScriptillä voidaan myös luoda keksejä, joilla voidaan tallettaa tietoja pidemmäksi aikaa esimerkiksi käyttäjän valitsemista asetuksista. (Negrino & Smith 2007, 6.)

JavaScriptissä on joitakin rajoitteita pääasiassa turvallisuussyistä. Esimerkiksi JavaScriptillä ei voi lukea asiakaspuolen tiedostoja, luoda tiedostoja palvelimelle tai vaikuttaa muihin avoimiin selainikunoihin. (Negrino & Smith 2007, 7.)

3.3 PHP-ohjelmointikieli

PHP (lyhenne sanoista PHP: Hypertext Preprocessor) on suunniteltu erityisesti web-sovelluskehitystä varten. PHP on tulkettava kieli, mikä tarkoittaa, että palvelin ajaa sen joka kerta, kun sivu ladataan. Selaimen kautta PHP-koodista näkyy ainoastaan sen antama tulostus. PHP-koodia voidaan lisätä suoraan HTML-dokumenttiin tai mihin tahansa palvelimella olevaan tekstimuodossa olevaan tiedostoon. (Heinisuo & Rauta 2007, 12–13.)

PHP:n avulla voidaan käyttää mm. monia tietokantoja varsin yksinkertaisilla komennoilla. Sen avulla voidaan lähettää kysely tietokantaan, kun sivu ladataan, ja tulostaa saatu hakutulos osaksi sivua. PHP sisältää kaikki ohjelmointikielille tyypilliset rakenteet. Se on oliopohjainen kieli, joten sillä rakennetut sovellukset voidaan toteuttaa haluttaessa luokkina. PHP on avoin ohjelmisto ja sen lähde koodi on saatavilla ilmaiseksi. (Heinisuo & Rauta 2007,13–14.)

3.4 SQL-kyselykieli

Nimi SQL tulee sanoista Structured Query Language eli strukturoitu kyselykieli. SQL-kielillä voidaan määritellä ja muuttaa tietokannan rakennetta. Sillä voidaan tehdä kyselyitä, jotka palauttavat haluttua tietoa tietokannasta. Sen avulla voidaan vaikuttaa tietokannan sisältöön lisäämällä, muuttamalla ja poistamalla sen tietoja. Lisäksi SQL kattaa myös seuraavat alueet: tapahtumankäsittely ohjaaminen, valtuuksien ja turvallisuuden hoito, upotettu SQL ja kohdistimen hallinta ja API-rajapinnat ohjelmointikieliin. (Hovi 2008, 14.)

SQL on ei-proseduraalinen kieli. Sillä ei kerrota, miten tietokantaoperaatiot tehdään, vain mitä halutaan tehtävän. SQL:ää voidaan käyttää vuorovaikutteisesti SQL-tuotteissa antamalla SQL-käskyjä erillisellä käyttöliittymällä ja saamalla sitten vastauksen käyttöliittymän eri ikkunaan. Tätä voidaan hyödyntää esimerkiksi, kun testataan käskyjä, ennen kuin ne upotetaan ohjelmiin. Upotetuksi SQL:ksi kutsutaan sitä, että SQL-käskyjä upotetaan ohjelmistokielen sisälle. Tällöin tietokannasta tulevat vastaukset siirretään ohjelmistokielen muuttujille. (Hovi 2008, 16.)

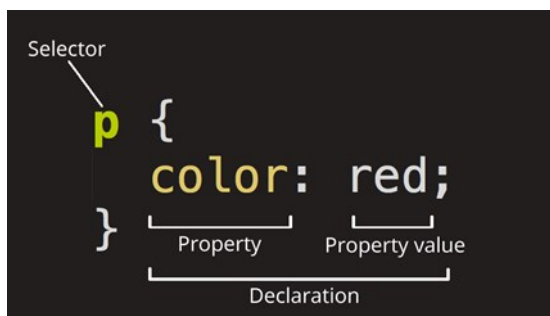
3.5 CSS

Cascading Style Sheets eli CSS on kieli, jota käytetään määrittelemään HTML- ja XML-dokumenttien ulkoasua näytöllä, paperilla tai muussa mediassa. Se on yksi keskeisimmistä kielistä, joita tarvitaan web-suunnittelussa. Eri internet-selaimet tukevat laajasti sen käyttöä. (MDN web docs 2019a.)

Ennen kuin CSS oli luotu, nettisivujen tyyli oli hyvin aikaa vievää ja kallista, koska tyyliä täytyi määritellä jokaiselle nettisivulle erikseen HTML-tiedostojen sisälle. World Wide Web Consortium

(W3C) kehitti CSS:n ongelman ratkaisemiseksi. Sen avulla sivujen tyylimäärittelyt saatiin ulkoistettua HTML-tiedostoista erilliseksi CSS-tiedostoksi. Nykypäivänä CSS:n käyttö on erittäin tärkeää, kun suunnitellaan sivuja, jotka osaavat mukautua käytettävän laitteen näyttön tai selainikkunan koon mukaan. (w3schools 2019b.)

CSS-syntaksi sisältää kaksi osaa: valitsijan (selector) ja määritelmän (declaration) (kuva 3). Valitsija määrittelee HTML-elementin, jota halutaan muokata. Valitsija voi olla myös HTML-tiedostossa käytetty id- tai luokka-arvo. Määritelmän tulee olla valitsijan perässä aaltosulkeiden sisällä. Määritelmä koostuu ominaisuudesta (property) ja ominaisuuden arvosta (property value), jotka on eroteltu kaksoispisteellä. Ominaisuuksia on yli 300 eri vaihtoehtoa ja niille on lähes loputon määrä eri arvoja. Määritelmiä voi olla aaltosulkeiden sisällä useampia ja ne on eroteltu puolipisteillä. Kuvan 3 esimerkki muuttaisi minkä tahansa HTML-tiedoston kaikkien <p>-elementtien(paragraph) värin punaiseksi. (MDN web docs 2019b.)



KUVA 3. Syntaksi esimerkki. (MDN web docs 2019b.)

Liitettäessä CSS-tiedostoa HTML-tiedostoon tarvitsee se ainoastaan linkittää HTML-tiedoston header-osaan, kuten alla olevassa esimerkissä. (w3schools 2019b.)

```
<head>  
< link rel="stylesheet" type="text/css" href="mystyle.css" >  
</head>
```

3.6 Kehykset ja lisätyökalut

3.6.1 Bootstrap

Bootstrap-ohjelmistokehys kehitettiin alun perin Twitteriä varten 2010 vuoden puolella välissä, jolloin se tunnettiin vielä nimellä Twitter Blueprint. Muutama kuukausi kehityksen aloituksesta Twitter järjesti ensimmäisen Hack Week -tapahtumansa, jossa projekti kasvoi räjähdysmäisesti kaiken tasoisten kehittäjien lähtiessä siihen mukaan. Se toimi sisäisenä tyylimallina yli vuoden, ennen kuin se julkaistiin julkiseen käyttöön vuonna 2011. (Bootstrap 2019.)

Bootstrapillä on ollut vuosien varrella yli 20 julkaisua, jotka sisältävät pari laajempaa uudelleenkirjoitusta. Bootstrap 2 -julkaisu lisäsi vaihtoehtoisia responsiivisiä toimintoja koko kehukseen. Bootstrap 3 -julkaisussa pyrittiin siihen, että responsiivisuus olisi kehyksessä vakiona, jotta se toimisi paremmin mobiililaitteiden kanssa. Bootstrap 4 -julkaisussa kehys kirjoitettiin uudelleen, jotta se voisi yhdistää Sass- ja CSS Flexbox -tekniikoita. Bootstrapiä ylläpidetään pienellä ryhmällä GitHubin kautta. Se on vuosien varrella kasvanut maailman suosituimmaksi front-end-kehykseksi ja avoimen lähdekoodin projektiksi. (Bootstrap 2019.)

3.6.2 Bootstrap Table

Bootstrap Table on laajennettu taulukkekehys, joka toimii monien yleisimmin käytettyjen CSS kirjastojen kanssa kuten, Bootstrap, Sematic UI, Bulma, Material Desing ja Foundation. Se on suunniteltu vähentämään sovelluskehitysaikaa. Se on kevyt, mutta sisältää paljon ominaisuuksia. (Bootstrap Table 2019.)

4 TOTEUTUS

4.1 Käyttöliittymä

Työn alussa tuli selvittää, millainen käyttöliittymäratkaisu olisi toimeksiantajalle paras ratkaisu. Aluksi tutustuttiin, millaisia työkaluja ja mahdollisuuksia olisi tarjolla, jos käyttöliittymä olisi oma ohjelmansa tai selainpohjainen.

Tutustumisen jälkeen päädyttiin selainpohjaiseen käyttöliittymään. Yrityksen toiveena oli, että käyttöliittymä näyttäisi hyvältä. Selainpohjaisessa käyttöliittymässä pystyttiin hyödyntämään valmiita taulukkokehyskiä ja CSS-kirjastoja, jotka auttoivat tämän tavoitteen saavuttamisessa. Ne myös auttoivat vähentämään sovelluskehittämiseen tarvittavaa aikaa. Lisäksi tekijällä oli henkilökohtaisesti enemmän kokemusta selainpohjaisten sovellusten tekemisestä. Myös se, että jokaisen sovellusta käyttävän työntekijän pitäisi asentaa erillinen ohjelma käyttääkseen tietokantaa, tuntui tarpeettomalta vaivalta.

Käyttöliittymän suunnittelu lähti liikkeelle pääsivusta. Pääsivun tuli sisältää luettelon kaikista tietokannassa olevista projekteista ja suodatustoiminnot. Alussa suunniteltiin paperille, mitä tietoja pääluettelossa tulisi esittää. Pian siirryttiin kuitenkin testailemaan ulkoasua selainympäristössä käyttäen HTML-elementtejä yhdessä Bootstrap-kirjaston kanssa. Kuvia käyttöliittymästä, joissa mainitut elementit näkyvät löytyy Tulokset-kappaleesta. Värien, fonttien ja logojen käytössä hyödynnettiin yrityksen sivujen ja muiden dokumenttien sisältöä. Luettelon taulukoinnissa käytettiin apuna Bootstrap Table -kehystä, josta löytyi valmiina monia hyödyllisiä ominaisuuksia. Näistä ominaisuuksista hyödynnettiin muun muassa tiedonjakamista useille sivuille, hakutoimintoa, tiedon ryhmitystä ja mahdollisuutta piilottaa haluttuja sarakkeita. Suodatustoiminnolle tehtiin oma valikko, jonka saa näkyville ja piiloon painiketta painamalla. Pääsivun kautta jokaista projektia voi tarkastella tarkemmin painamalla Tiedot-painiketta, joka avaa ikkunan, josta näkee kaikki projektin tiedot tarkasti, kuten kuka on työskennellyt projektissa ja missä roolissa. Tämän ikkunan takaa löytyy myös Muokkaa-painike, joka vie käyttäjän sivulle, jossa projektin tietoja voi muokata.

Seuraavana vuorossa oli Henkilöhaku-sivu. Sivun tarkoituksena oli tuottaa tuloste, joka voidaan suoraan liittää työntekijän CV:hen tai esittää tarjoustoiminnan yhteydessä. Tällä sivulla on valintalaatikko, joka sisältää kaikkien tietokannassa olevien työntekijöiden nimet. Kun laatikosta valitsee nimen, sivulle tulee taulukko, joka sisältää kaikki projektit, missä henkilö on ollut mukana, hyvin samanlaisella tyyllillä kuin pääsivullakin. Suunniteltaessa, mitä tietoja tällä sivulla näytetään ja millaisessa muodossa, käytettiin apuna yrityksen aikaisempia henkilöesitteitä. Myös tällä sivulla käyttäjä voi suodattaa hakutuloksia painikkeen takaa löytyvien suodatustoimintojen avulla. Myöhemmin tälle ja pääsivulle lisättiin ominaisuus, jolla voi tulostaa sivusta A4-kokoisen pdf- tai paperitulosteen. Tässä ominaisuudessa hyödynnettiin CSS:stä löytyvää ominaisuutta, jolla voidaan muokata, minäkälaisena sivun tuloste näkyy. Näin tulostetta ei tarvitse erikseen muokata käsin.

Sen jälkeen, kun saatiin selvitettyä, mitä tietokantaa projektissa käytetään, täytyi sivulle tehdä toiminnot, joilla pystyy lisäämään tietokantaan uuden projektin sekä hallitsemaan ja lisäämään uusia käyttäjiä. Käyttäjien lisäämistä varten täytyi sivuille myös lisätä kirjautumissivu. Kaikissa lomakkeissa, joissa lisätään tai muokataan tietokannan sisältöä, hyödynsin Bootstrapin komponentteja. Kaikilla sivuilla, jotka listaavat tietokannan tietoja, hyödynnettiin Bootstrap Tablea. Käyttäjillä voi olla eritasoisia käyttäjätunnuksia, jotka rajaavat, mitä sivuja he voivat nähdä ja mitä toimintoja he pystyvät käyttämään. Myös tietoturvallisuus on otettu suunnittelussa huomioon.

4.2 Tietokanta

Tietokantaa valittaessa tutustuttiin, millaisia palveluja yrityksellä oli ennestään käytössä ja olisiko niistä apua projektissa. Potentiaalisimmalta palvelulta vaikutti ValueFrame-toiminnanohjausjärjestelmä, mihin yritys kirjasi jo joitakin tietoja jokaisesta projektista. ValueFramen tarjonnasta löytyi myös lisäpalveluna tietovarasto-palvelu, joka tarjoaa pääsyn ValueFrameen tallennettuun dataan. Tarkemman tutustumisen jälkeen kävi kuitenkin ilmi, että kaikkia referenssilistaan tarvittavia tietoja ei saisi ValueFrameen.

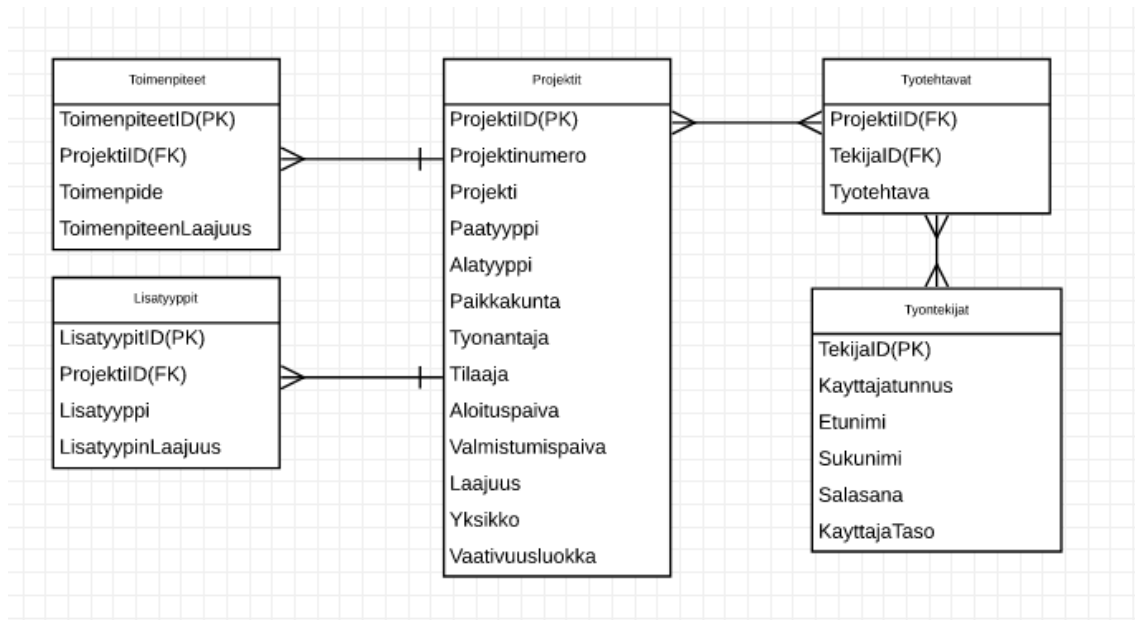
Projektissa päädyttiin käyttämään MySQL-tietokantaa, koska se oli tekijälle jokseenkin tuttu ja sillä pystyi hyvin tekemään kaikki tarvittavat ominaisuudet. Tietokannan rakenteen suunnittelu aloitettiin jakamalla vanhan Excel-tilin tiedot erillisiin tauluihin. Tarvittavia tietoja ja tauluja lisättiin uusia ominaisuuksia varten, kuten käyttäjätunnuksia varten. Taulujen välille tehtiin sopivat pää- ja vii-

teavaimet. Projektia tehdessä tietokannan rakenteeseen tuli pieniä muutoksia, kun yritys keksi uusia ominaisuuksia. Tietokanta ja käyttöliittymä toimivat yrityksen sisäisessä verkossa yrityksen omalla palvelimellaan.

5 TULOKSET

5.1 Tietokannan rakenne

Tietokannan suurin taulu Projektit sisältää suurimman osan yrityksen aikaisemmin käyttämän Excel-taulukon sisällöstä (kuva 4). Lisäksi taulukkoon on lisätty joitakin uusia sarakkeita yrityksen pyynnöstä. Taulukon perusavaimena toimi aluksi Projektinumero, mutta vaihdoin sen myöhemmin automaattisesti numeroituun ProjektiID:hen, koska yritys toivoi, että taulukkoon voidaan lisätä projekteja myös samalla projektinumerolla, jos halutaan kirjata esimerkiksi osaprojekti.



KUVA 4. Tietokannan rakenne

Aikaisemmassa Excel-taulukossa oli sarakkeita nimeltä Toimenpide1 ja Toimenpide2. Tein tietokantaan näille sarakkeille oman taulun nimeltä "Toimenpiteet". Toimenpiteet on yhdistetty Projektit-tauluun isä-lapsiyhteydellä käyttäen ProjektiID:tä viiteavaimena. Näin yritys voi lisätä niin monta toimenpidettä projektille kuin on tarpeen. Lisäksi tauluun pystyy kirjaamaan toimenpiteen laajuuden m²-yksikkönä.

Projektit on jaettu yleensä päätyypin ja alatyypin mukaan. Esimerkiksi yksi projekti voi olla päätyypiltään julkinen rakennus ja alatyypiltään sairaala. Jotkin projektit voivat kuitenkin kuulua useaan eri alatyypin. Esimerkiksi projekti voi olla julkinen rakennus ja alatyyppeihin voi kuulua koulu, päiväkotit ja urheilurakentaminen. Tätä varten loin tietokantaan taulun Lisätyypit, joka on yhdistetty

Projektit-tauluun isä-lapsiyhteydellä käyttäen ProjektiID:tä viiteavaimena. Näin yritys voi kirjata tällaisen rakennuksen tietokantaan merkkiaamalla alatyypiksi monitoimirakennuksen ja lisäämällä sille tarvittavan määrän lisätyyppejä. Lisätyyppeihin voi myös kirjata sen laajuuden m²-yksikkönä.

Työntekijät-taulu sisältää tietoja yrityksen työntekijöistä. Työntekijät-taulun perusavain on TekijäID. Lisäksi jokaisella työntekijällä on käyttäjätunnus, etunimi, sukunimi, salasana sekä käyttäjätaso. Salasanoja ei säilytetä tietokannassa selkokielisinä. Käyttäjätasoa käytetään rajoittamaan käyttäjien oikeuksia käyttöliittymässä.

Jokaisella projektilla voi olla monta työntekijää, jotka toimivat erilaisissa tehtävissä. Jokainen työntekijä voi toimia useassa eri projektissa eri työtehtävässä. Tätä varten tietokantaan tehtiin välitaulu nimeltä Työtehtävät. Taulussa on kaksi viiteavainta. ProjektiID viittaa Projektit-taulun tietoihin ja TekijäID viittaa Työntekijät-taulun tietoihin. Lisäksi taulussa on sarake työtehtävälle.

5.2 Käyttöliittymä

Käyttöliittymän ensimmäinen sivu on sisäänkirjautumissivu. Tällä sivulla on kenttä käyttäjätunnukselle ja salasanalle sekä painike kirjautumista varten. Vaihtoehtoisesti käyttäjä voi painaa Enter-näppäintä kirjautuessaan sisään. JavaScript-osio poimii käyttäjän syöttämät arvot ja välittää ne PHP-osioon, joka tarkistaa ne. Jos käyttäjä syöttää väärän käyttäjätunnuksen tai salasanan näytetään siitä hänelle virheilmoitus. Jos syötetyt arvot täsmäävät ja käyttäjän tunnuksilla on oikeus kirjautua sivulle, ohjataan käyttäjä pääsivulle. Tietoturvallisuus on otettu huomioon sivustolla ja jokainen sivu sivustolla tarkistaa, että käyttäjä on tullut kirjautumissivun kautta.

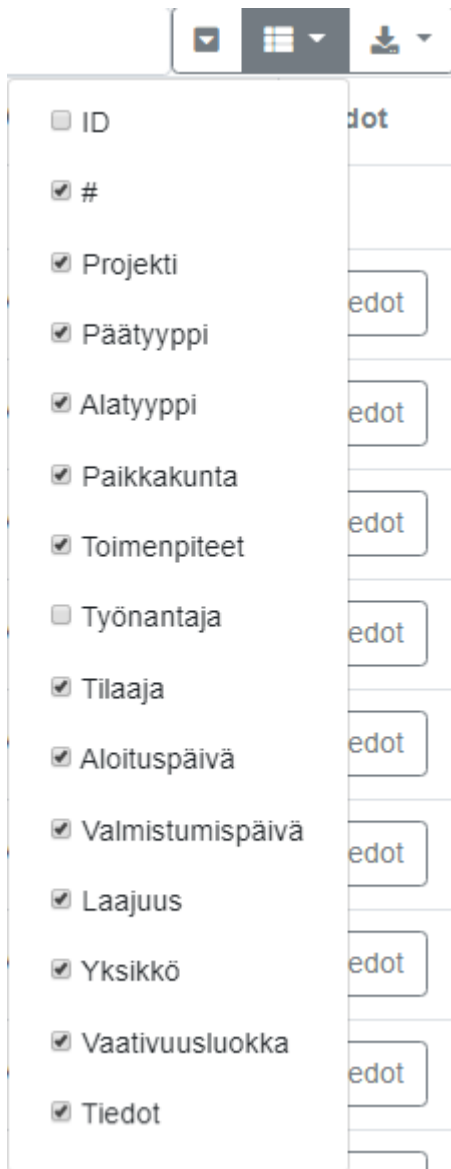
Kun pääsivu ladataan käynnistyvät latauksen yhteyteen liitetyt funktiot JavaScriptissä. Ensimmäisenä haetaan kaikki tiedot tietokannasta koskien projekteja. JavaScript tekee AJAX XMLHttpRequest kyselyn PHP-osioon, joka sisältää SQL-kyselyn tietokantaan. Vastauksena PHP-osio palauttaa JSON-tilukon, joka säilötään JavaScript-muuttujaan. Tällainen kysely tehdään jokaiseen tietokannan sisältämään tauluun ja saadut tiedot säilötään omaan JavaScript-muuttujaan. Kun tiedot on saatu, yhdistetään erillisissä JavaScript-muuttujissa olevat tiedot yhdeksi suureksi JSON-tilukoksi, jota kutsutaan nimellä masterTable. Tämä taulukko sisältää olion jokaisesta projektista ja jokainen olio sisältää kaiken tarvittavan tiedon kyseisestä projektista, kuten perustiedot tietokannan taulukosta Projektit, kaikki projektissa toimineet työntekijät ja heidän työtehtävänsä ja

niin edelleen. Kun masterTable on valmis, voidaan siitä poimia käyttäjälle näytettävässä taulukossa käytettävät tiedot, muokata niiden esitystapaa tarvittaessa ja luoda siitä Bootstrap Table-tila, jonka käyttäjä näkee sivulla (kuva 5). Esitystapaa muokataan esimerkiksi lisäämällä lukuihin tuhat-erottimet ja muokkaamalla päivämäärä haluttuun formaattiin.

| # | Projekti | Päätyyppi | Alatyyppi | Paikkakunta | Toimengiteet | Tilaja | Alloituspäivä | Valmistumispäivä | Laajuus | Yksikkö | Vaativuusluokka | Tiedot |
|-----------------------------------|----------|-------------------------|-----------------------------------|-------------|--|--------|---------------|------------------|---------|----------------|-----------------|--------|
| 123 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Peruskorjaus | | 2011-1-1 | 2012-1-1 | 4 700 | m ² | Vaativa | Tiedot |
| 125 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Peruskorjaus, Laajenus | | 2010-1-1 | 2011-1-1 | 3 500 | m ² | Vaativa | Tiedot |
| 133 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Korjaus | | 2008-1-1 | 2009-1-1 | 700 | m ² | Vaativa | Tiedot |
| 146 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Peruskorjaus | | 2011-1-1 | 2012-1-1 | 6 600 | m ² | Vaativa | Tiedot |
| 157 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Peruskorjaus, Uudisrakennus, Tärjousvälheen viitesuunnittelu | | 2009-1-1 | 2010-1-1 | 15 900 | m ² | Vaativa | Tiedot |
| 158 | | Julkiset rakennukset | Koulut ja päiväkodit | Oulu | Uudisrakennus | | 2014-1-1 | 2015-1-1 | 16 000 | m ² | Vaativa | Tiedot |
| Toimistorakennukset ja toimitilat | | | | | | | | | | | | |
| 19 | | Tuotanto- ja toimitilat | Toimistorakennukset ja toimitilat | Oulu | Muutos työ | | 2005-1-1 | 2006-1-1 | 1 200 | m ² | Tavanomainen | Tiedot |
| 24 | | Tuotanto- ja toimitilat | Toimistorakennukset ja toimitilat | Oulu | Tilamuutos, Sisustus | | 2005-1-1 | 2007-1-1 | 300 | m ² | Tavanomainen | Tiedot |
| 39 | | Tuotanto- ja toimitilat | Toimistorakennukset ja toimitilat | Oulu | Korjaus | | 2005-1-1 | 2007-1-1 | 2 500 | m ² | Tavanomainen | Tiedot |
| 48 | | Tuotanto- ja toimitilat | Toimistorakennukset ja toimitilat | Oulu | Tilamuutos | | 2005-1-1 | 2007-1-1 | 400 | m ² | Tavanomainen | Tiedot |

KUVA 5. Pääsivun näkymä. Projektinimet ja tilaajat sensuroitu.

Bootstrap Table jakaa tiedot automaattisesti sivuille, joita käyttäjä voi selata alhaalta oikealta löytyvien näppäinten avulla. Käyttäjä voi myös halutessaan päättää, kuinka monta tulosta sivulla näytetään alhaalta vasemmalta löytyvän valikon avulla. Bootstrap Tablesta löytyy myös ominaisuus, jolla tiedot voidaan järjestää numero- tai aakkosjärjestykseen halutun sarakkeen mukaan. Taulukon yläpuolella vasemmalla on infoteksti, joka kertoo, kuinka monta tulosta sivulla näytetään ja mikä on niiden yhteispinta-ala. Sulkeissa näkyy eroteltuna projekteissa käytetyt erilaiset yksiköt. Tämä on toteutettu itse tehdyllä funktiolla. Ylhäältä oikealta löytyy kolme neliönmuotoista painiketta Bootstrap Table -ominaisuuksille. Ensimmäisellä painikkeella vasemmalta käyttäjä voi ladata taulukosta Excel-version. Keskimmäisellä painikkeella käyttäjä voi valita, mitä sarakkeita taulukossa näytetään (kuva 6). ID- ja Työnantaja-sarakkeet ovat piilotettuina vakionäkymässä. Viimeistä painiketta painamalla tietojen jako sivuiksi poistuu käytöstä ja tiedot listataan yhdeksi pitkäksi sivuksi.



KUVA 6. Bootstrap Table -ominaisuuksien painikkeet sarakevalikko aukaistuna.

Näiden painikkeiden vierestä löytyy pikahaku-kenttä, joka tulee suoraan Bootstrap Tablen -ominaisuuksista. Kun käyttäjä kirjoittaa kenttään, Bootstrap Table suodattaa automaattisesti hakua vastaavat tulokset taulukkoon.

Hakukentän vasemmalla puolella on kaksi itse tehtyä painiketta. Vasemmanpuoleisin painike on Tulosta-painike, jonka avulla käyttäjä voi tulostaa taulusta A4-kokoisen paperi- tai pdf-version. Tämä on toteutettu käyttämällä CSS -ominaisuutta `media print`, jolla voi muokata tulostuksen kuvasuhdetta, tekstin kokoa ja piilottaa turhia elementtejä kuten navigointivalikko ja painikkeet näkyvistä. Oikeanpuoleinen painike tuo esiin Tarkempi haku -valikon (kuva 7). Valikon alaosasta löytyy kuusi eri valintalaatikkoa eri sarakkeita kohden. Jokainen laatikko sisältää kaikki uniikit arvot mitä

kyseisellä sarakkeella on. Arvot on suodatettu laatikoihin masterTable-taulukosta. Valikosta löytyy myös vaihtoehdot hakea tietoja tiettyjen päivämäärien väliltä, käyttäjän määrittämän vuosimäärän takaa, tietyltä projektin laajuus haarukalta ja tietyltä toimenpiteen laajuus haarukalta. Toimenpiteen laajuus haarukka haku on käytössä vain, jos Toimenpide-valintalaatikkoon on valittu jokin tietty toimenpide. Kaikkia näitä hakuvaihtoehtoja ohjataan saman funktion avulla, joka käynnistyy, kun jokin arvo valikossa muuttuu. Funktio tarkistaa, mitä hakuvaihtoehtoja käyttäjä on valinnut ja suodattaa halutut tiedot masterTable-taulukosta. Käyttäjä voi valita useita eri hakuvaihtoehtoja ja myös Pikahaku-kenttä toimii yhdessä Tarkempi haku -ehtojen kanssa. Bootstrap Tablessa on ominaisuus, jolla voidaan luoda taulukkoon väliotsikoita helpottamaan sen luettavuutta. Ryhmitys-valintalaatikosta käyttäjä voi valita luodaanko nämä väliotsikot alatyypin vai päätyyppisarakeen mukaan. Käyttäjä voi myös ottaa väliotsikoinnin pois käytöstä valintalaatikon alla olevasta valintaruudusta. Käyttäjä voi myös vaihtaa formaattia, jolla päivämäärä esitetään taulukossa.

KUVA 7. Tarkempi haku -valikko.

Jokaisella projektilla on oma Tiedot-painike, joka sijaitsee Tiedot-sarakkeessa. Kun käyttäjä painaa painiketta, tulee Bootstrap modale esille. Bootstrap modale on ponnahdusikkunamainen elementti, joka ei tarvitse omaa selainikkunaa. Ikkunan koon voi määrittää CSS-arvoilla ja sen sisälle voidaan sijoittaa haluttuja HTML-elementtejä. Ikkuna sijoittuu suoraan pääsivun päälle ja sen ulkopuolinen sisältö on himmennetty. Tiedot-painike on toteutettu siten, että painikkeen painallus välittää projektin ID:een funktiolle, joka suodattaa tarvittavat tiedot masterTable-taulukosta. Tiedot-modale sisältää kaiken tiedon projektista. Sen alkuosaan on listattu perustiedot ja alaosasta löytyy kolme taulukkoa. Ensimmäinen on taulukko työntekijöistä, jotka ovat työskennelleet projektissa. Siinä on sarakkeet etunimelle, sukunimelle sekä työtehtävälle. Toinen taulukko on toimenpiteistä. Siinä on sarakkeet toimenpiteille ja niiden laajuuksille. Kolmas taulukko on lisätyypeille. Siinä on sarakkeet lisätyypeille ja niiden laajuuksille. Modalen alaosasta löytyy kaksi painiketta. Toinen on modalen sulkemista varten. Käyttäjä voi myös sulkea modalen yläosasta löytyvästä rastikuvakkeesta tai klikkaamalla modalen ulkopuolelle. Toinen painike on projektin muokkaamista varten. Kun käyttäjä painaa painiketta, ohjataan hänet uudelle sivulle.

ProjektiID siirretään uudelle muokkaussivulle URL-osoitteen avulla. Kun sivu ladataan, haetaan tietokannasta ProjektiID:tä vastaavat tiedot, sekä kaikki tiedot, joita tarvitaan sivulla olevien valintalaatikoiden generoimiseen. Sivun HTML-elementtien yhteydessä hyödynnetään Bootstrap-kirjastoja. Muokkaus-sivu on lomake, jossa on teksti- tai numerokentät projektin nimelle, työnantajalle, tilaajalle ja laajuudelle. Laajuuden yksikölle ja vaativuusluokalle on valintalaatikot, joiden arvot ovat staattisia. Aloitus- ja valmistumispäiville on kentät, joihin voi asettaa päivämäärät pienen kalenterin avulla. Päätyypille ja paikkakunnalle on valintalaatikot, jotka sisältävät kaikki uniikit arvot, jotka löytyvät Projektit-taulun vastaavista sarakkeista tietokannassa. Alatyypeille on myös valintalaatikko, mutta se sisältää vain uniikkeja arvoja Projektit-taulusta Alatyyppi-sarakkeesta, joilla on myös valittuna oleva päätyyppi.

Lisätyypeille, toimenpiteille ja työntekijöille on varattuna omat alueet. Näiltä alueilta löytyy Lisää uusi -painike. Lisätyypit-alueella painiketta painamalla generoituu alueelle uusi rivi, joka sisältää valintalaatikon Lisätyyppi-arvolle, numerokentän sen laajuutta varten ja Poista-painikkeen (kuva 8). Valintalaatikko sisältää uniikkeja arvoja Lisätyypit-taulusta Lisätyyppi-sarakkeelta ja Projektit-taulusta Alatyyppi-sarakkeelta, joita valittuna olevalla päätyypillä on. Toimenpiteet-alueella painike generoi uuden rivin, jolla on valintalaatikko toimenpiteelle, numerokentän sen laajuudelle ja Poista-painike. Valintalaatikko sisältää uniikkeja arvoja Toimenpiteet-taulusta Toimenpide-sarakkeelta, joita valitulla päätyypillä on. Työntekijät-alueella painike generoi rivin, jolla on valintalaatikko työntekijälle ja työtehtävälle sekä Poista-painike. Työntekijä-valintalaatikossa on etunimi + sukunimiyhdistelmä kaikista työntekijöistä, jotka ovat Työntekijät-taulussa. Työtehtävä-valintalaatikossa on kaikki uniikit arvot, jotka ovat Työtehtävät-taulussa Työtehtävä-sarakkeella. Kaikki valintalaatikot, joiden arvot eivät ole staattisia, sisältävät myös arvon "Muu". Poikkeuksena on Työntekijä-valintalaatikko. Kaikkiin kenttiin ja valintalaatikoihin on vakiona asetettuna arvo, joka vastaavalla ProjektiID:llä on. Lomakkeen alaosasta löytyy Tallenna ja Poista -Painikkeet.

Lisätyypit

| Lisätyyppi | Lisätyypin laajuus m ² | |
|------------|-----------------------------------|--------|
| Koulu | 20000 | Poista |
| Päiväkoti | 10000 | Poista |
| Lisää uusi | | |

KUVA 8. Lisätyypit-alue, jolla on kaksi riviä.

Kahdella korkeimmalla käyttäjätasolla pystyy muokkaamaan ja poistamaan projektin tietoja vapaasti. Poikkeuksena on projektinumero, jota ei voi vaihtaa jälkikäteen. Perustyöntekijän tunnuk-silla pystyy vain lisäämään tai poistamaan oman työtehtävän. Jos johonkin valintalaatikkoon valitsee arvon ”Muu”, sivulle avautuu ponnahdusikkuna, johon voi syöttää uuden arvon. Valintalaatikot on suunniteltu näin, jotta tietokannan arvot pysyvät mahdollisimman yhtenäisinä, eikä esimerkiksi yhdelle alatyypille synny monta eri arvoa, jotka ovat kirjoitettu himan eri tavoilla. Jos käyttäjä painaa Poista-painiketta Lisätyypit-, Toimenpiteet- tai Työntekijät-alueilla olevalta riviltä, poistetaan rivi välittömästi lomakkeesta ja tietokannasta. Jos käyttäjä muuttaa jotain arvoa lomakkeessa, laukeaa muutoksen yhteyteen liitetty JavaScript-funktio. Funktio merkkää boolean-muuttujien avulla mihin tietokannan tauluun muutos kuuluu. Kun käyttäjä painaa Tallenna-painiketta toinen funktio tarkistaa, että lomakkeessa ei ole tyhjiä arvoja ja näyttää tarvittaessa virheilmoituksen ponnahdusikkunassa. Jos kaikissa kentissä on arvo funktio tarkistaa, mihin tauluihin tulee tehdä muutoksia boolean-muuttujien avulla, ja välittää muutetut arvot PHP-osioon, joka päivittää ne tietokantaan. Tämän jälkeen käyttäjälle näytetään ponnahdusikkunassa ilmoitus onnistumisesta tai virheestä.

Kahdella korkeimmalla käyttäjätasolla löytyy sivuston navigointipalkista Lisää projekti -sivu. Tämä sivu on samanlainen lomake kuin Muokkaa-sivukin, mutta sen kenttiin ei ole valmiiksi asetettuja arvoja, poikkeuksina projektinumero, johon ehdotetaan seuraavaa vapaata numeroa ja työnantaja, johon ehdotetaan yrityksen nimeä. Valintalaatikot on täytetty samalla tavalla kuin muokkaus sivulakin.

Henkilöhaku-sivulla voidaan hakea projekteja, joissa tietty työntekijä on työskennellyt. Aluksi, kun sivu avataan, käyttäjä voi valita valintalaatikosta työntekijän nimen. Valintalaatikosta löytyvät kaikkien työntekijöiden nimet, joilla on tietokannassa merkintä, että he ovat työskennelleet jossain projektissa. Kun käyttäjä valitsee nimen, listataan projektit, joissa työntekijä on työskennellyt, Bootstrap Tableen. Tämän sivun ulkoasu ja toiminnot ovat hyvin samanlaisia kuin pääsivullakin, pienillä muutoksilla. Uutena sarakkeena taulusta löytyy työtehtävä. Oletus-näkymän sarakkeet on valittu ja muotoiltu siten, että ne sopivat suoraan henkilöesitteen tulostukseen. Tulostuksen yläosasta löytyy myös tekstit henkilöesite, henkilön nimi sekä päivämäärä. Alaosasta löytyy yrityksen nimi. Tarkempi haku valikosta löytyy tällä sivulla myös Valintalaatikko-työtehtävälle.

Yhteenvetosivu on sivu, joka näkyy kahdelle korkeimmalle käyttäjätasolle. Se sisältää Bootstrap Table-taulun, jossa on sarakkeet ala- ja lisätyypeille, hankemäärille ja niiden yhteispinta-alalle. Kun sivu ladataan, kaikki tarvittava tieto haetaan tietokannasta ja JavaScript funktio erittelee ja laskee

tarvittavat tiedot, jonka jälkeen se asettaa ne taulukkoon. Sivu toimii nopeana infona tietokannan sisällöstä.

Hallitse käyttäjä-sivu on sivu, joka näkyy vain korkeimmalle käyttäjätasolle. Sivulla on Bootstrap Table-taulu, jossa on sarakkeet etunimi, sukunimi ja muokkaa. Taulukkoon on listattu kaikkien työntekijöiden nimet ja Muokkaa-sarakkeesta löytyy Muokkaa-painike. Taulukon yläosassa on myös hakupalkki. Lisäksi yläosasta löytyy Lisää työntekijä -painike. Jos käyttäjä painaa Lisää työntekijä -painiketta siirretään käyttäjä sivulle, jossa on lomake uuden työntekijän lisäämistä varten. Lomakkeessa on tekstikentät etunimelle, sukunimelle, käyttäjätunnukselle, salasanalle, salasanan toistamiselle sekä valintalaatikko käyttäjätasolle (kuva 9). Lomakkeen alaosasta löytyy Tallenna-painike. Kun käyttäjä painaa Tallenna-painiketta, funktio tarkistaa, että salasanat täsmäävät ja että kaikki kentät ovat täytetty. Jos tiedoissa on virhe, näytetään käyttäjälle virheilmoitus. Jos tiedot ovat oikein, ohjautuu sivu takaisin Hallitse käyttäjä-sivulle ja uusi tunnus löytyy taulusta. Kun käyttäjä painaa Muokkaa-painiketta siirretään käyttäjä sivulle, jossa käyttäjä voi muokata kyseisen rivin käyttäjää. Sivulla on samanlainen lomake kuin Työntekijän lisäys -sivullakin ilman salasanakenttiä. Lomakkeen alaosassa on lisäksi uusi painike, "Resetoi salasana". KäyttäjätID siirretään sivulle URL-osoitteen avulla. Käyttäjän tiedot haetaan tietokannasta ja asetetaan kenttiin. Käyttäjä voi muokata työntekijän tietoja ja tallettaa ne tietokantaan, jonka jälkeen käyttäjälle näytetään ponnahdusikkuna, jossa on viesti joko onnistuneesta päivittämisestä tai virheilmoitus. Jos käyttäjä painaa Resetoi salasana -painiketta asetetaan työntekijän uudeksi salasanaksi "Resetoitu". Työntekijää ei voi poistaa, koska tällöin jouduttaisiin poistamaan kaikki rivit lapsitauluista joihin työntekijä viittaa. Käyttäjätason voi kuitenkin vaihtaa vanhaksi työntekijäksi, jolloin tunnuksilla ei pääse enää kirjautumaan sisään.

Etunimi

Sukunimi

Käyttäjätunnus

Salasana

Toista salasana

Käyttäjätaso

KUVA 9. Käyttäjänlisäys sivun lomake

Navigointipalkin oikeassa reunassa on käyttäjän nimi ja pieni nuoli symboli. Nimeä painamalla avautuu pudotusvalikko, jossa on painikkeet Omat tiedot ja Kirjaudu ulos. Omista tiedoista käyttäjä näkee oman etu- ja sukunimen sekä käyttäjätunnuksen. Tietojen alapuolella on painikkeet Muokkaa nimeä sekä Vaihda salasana. Jos käyttäjä painaa Muokkaa nimeä -painiketta, ohjataan hänet lomakkeeseen, jossa hän voi vaihtaa etu- tai sukunimensä esimerkiksi naimisiin menon johdosta. Lomakkeen alaosassa on Tallenna-painike, jota painamalla tiedot päivittyvät tietokantaan. Vaihda salasanaa-painiketta painamalla käyttäjä ohjataan lomakkeeseen, jossa on kenttä vanhalle salasalle ja kaksi kenttää uudelle salasalle. Käyttäjän täytyy syöttää vanha salasana oikein sekä uusi salasana kaksi kertaa ja painaa Talleta-painiketta, jotta salasana vaihtuu, jonka jälkeen hänelle näytetään viesti "salasana vaihdettu onnistuneesti". Jos jokin syötetyistä tiedoista on väärin, näytetään siitä virheilmoitus.

6 POHDINTA

Työn tarkoituksena oli luoda toimiva tietokanta ja sille käyttöliittymä yrityksen referenssiluetteloa varten. Työ oli tarkoitus saada valmiiksi 2019 kevään aikana. Uudessa tietokannassa on enemmän sarakkeita ja tauluja kuin vanhassa Excel taulukossa, minkä takia vanhoja tietoja täytyy muokata, ennen kuin ne voidaan siirtää uuteen tietokantaan. Tämän takia järjestelmää ei ole vielä otettu käyttöön. Tein kuitenkin yrityksen sisäverkossa toimivan version tietokannasta ja käyttöliittymästä valmiiksi kevään 2019 aikana ja se on odottamassa, että tiedot saadaan siirrettyä tietokantaan. Tietojen siirtoakin on jo kokeiltu onnistuneesti ensimmäisen 150 projektin tiedoilla, jotka yritys on saanut muokattua.

Ohjelmaa luultavasti joutuu hieman jatkokehittämään käyttöönoton yhteydessä ja sen jälkeen. Käyttöliittymästä tulee varmasti vielä löytymään joitakin virheitä, joita en ole onnistunut omalla testauksellani löytämään, ja ne tulee korjata. Yrityksellä tulee varmasti vielä ideoita, joilla käyttöliittymän käytettävyyttä voidaan parantaa, kun he itse pääsevät käyttämään sitä. Olen kuitenkin koittanut suunnitella käyttöliittymän ja tietokannan siten, että tulevaisuudessa yritys voi itsenäisesti ylläpitää järjestelmää.

Työn toteutus sujui kohtuullisen vaivattomasti. Olin käyttänyt web-ohjelmointikieliä ja tietokantoja jo aikaisemmissa kouluprojekteissa ja kursseilla. Suurimpana haasteena oli suunnitella tietokannan rakenne. En ole aikaisemmin joutunut suunnittelemaan näin laajaa tietokantaa, jossa on myös isälapsiyhteyksiä. Aiheesta oli kuitenkin saatavilla hyvin opintomateriaalia. Uutena kehyksenä käytin Bootstrap-kirjastoa. Halusin käyttää sitä työssä, koska olin kuullut, että se on yksi yleisimmin käytetyistä, front-end -kirjastoista ja ajattelin, että sen osaamisesta on hyötyä tulevaisuudessa. Kirjaston käyttäminen osoittautui helpoksi ja sillä oli hyvä dokumentaatio. Työläintä työssä oli erilaisten suodatustoimintojen koodaaminen, koska niitä oli niin monta ja yritys keksi niitä lisää työn edetessä. Kirjoitusosuuden tekeminen venyi työssä ja se saatiin valmiiksi vasta 2019 syyslukukaudella. Olen tyytyväinen siitä, mitä sain aikaiseksi ja odotan innolla, että järjestelmä päästään ottamaan käyttöön. Myös yritys oli tyytyväinen järjestelmään, jonka esittelin keväällä 2019. Opin paljon uutta tietokannoista ja web-kehittämisestä, josta on varmasti hyötyä tulevaisuudessa.

LÄHTEET

Bootstrap. 2019. About. Viitattu 16.5.2019, <https://getbootstrap.com/docs/4.3/about/overview/>.

Bootstrap Table. 2019. Viitattu 16.5.2019, <https://bootstrap-table.com/>.

Heinisuo, Rami – Rauta, Ilkka 2007.PHP ja MySQL Tietokantapohjaiset verkkopalvelut. Helsinki: Talentum

Hovi, Ari 2008. SQL-opas. Jyväskylä: Docendo Finland Oy

Hovi, Ari - Huotari, Jouni - Lahdenmäki, Tapio 2005. Tietokantojen suunnittelu ja indeksointi. Porvoo: Docendo Finland Oy

Korpela, Jukka - Linjama, Tero 2005. Web-suunnittelu. Porvoo: Docendo Finland Oy

Linja Arkkitehdit. 2019. Yritys. Viitattu 20.5.2019, <https://www.linja-arkkitehdit.fi/yritys/>.

MDN web docs. 2019a. CSS: Cascading Style Sheets. Viitattu 23.4.2019, <https://developer.mozilla.org/en-US/docs/Web/CSS>.

MDN web docs. 2019b. CSS syntax. Viitattu 23.4.2019, https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Syntax.

MDN web docs. 2019c. HTML basics. Viitattu 23.04.2019, https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics.

Negrino, Tom - Smith, Dori 2007. JavaScript – Tehokas hallinta. Suom. Mikko Kamppila. Helsinki: Readme.fi

Oracle 2019. MySQL. Viitattu 20.5.2019, <https://www.oracle.com/mysql/>.

w3schools. 2019a. CSS How To.... Viitattu 23.4.2019,
https://www.w3schools.com/css/css_howto.asp.

w3schools. 2019b. CSS Introduction. Viitattu 23.4.2019, https://www.w3schools.com/css/css_intro.asp.