



Mobiilisivuston MikamaticMobiili kehittäminen

Annemari Mustonen

2020 Laurea



Laurea-ammattikorkeakoulu

Mobiilisivuston MikamaticMobiili kehittäminen

Annemari Mustonen

Tietojenkäsittely

Opinnäytetyö

Joulukuu, 2020

Annemari Mustonen

Mobiilisivuston MikamaticMobiili kehittäminenVuosi 2020 Sivumäärä 30

Tässä opinnäytetyössä keskityin kehittämään henkilö- ja yritystietoja yrityksen omasta Oracle-tietokannasta hakevaa ja niitä päivittävää mobiilisivustoa espoolaiselle Mikamatic Oy:lle. Työ aloitettiin tekemällä kartoitus yrityksen nykyisestä ohjelmasta, Yritysrekisteristä josta nämä toiminnallisuudet jo löytyvät. Suunnittelu aloitettiin MarvelApp- ohjelmalla, jolla se oli tarkoitus tehdä. Työn edetessä kuitenkin suunnittelu siirtyi pikku hiljaa Visual Studio Code - ohjelman puolelle, jossa myös varsinainen ohjelmointityö tapahtui. Aluksi itse ohjelmointiin käytettiin Ionic Frameworkkiä, mutta eteen tulleiden tietokantayhteyksiin liittyvien ongelmien takia se jätettiin pois jo alkuvaiheessa ja ohjelmointi suoritettiin täysin käyttäen HTML- ja Typescript- ohjelmointikieliä.

Itse sivusto tehtiin muista Mikamatic Oy:n ohjelmista tuttua Single Page Application - tyyliä käyttäen jolloin pääsivu pysyy aina samana, mutta sen sisältö muuttuu käyttäjän toimien mukaan. Tämän yhden sivun sisälle rakennettiin frame-komponenteista sekä henkilötiedot että yritystiedot hakevat taulukot, joissa näkyvät perustiedot kustakin henkilöstä tai yrityksestä. Sitten tehtiin sivut yksittäisen henkilön tai yrityksen laajempien tietojen näyttämiseksi. Tähän sivulle käyttäjä pääsee valitsemalla pääsivun taulukosta kyseisen henkilön tai yrityksen ja sen jälkeen klikkaamalla Katso-painiketta. Erikseen ohjelmoitiin myös kirjautumissivu lähes valmiiksi, mutta tällä hetkellä kirjautumistietoja ei voi erikseen antaa vaan sivusto tekee kirjautumisen automaattisesti.

Sivusto toimii tällä hetkellä ainoastaan Mikamatic Oy:n oman, hyvin vpn-yhteyksillä, virustorjuntaohjelmilla sekä palomureilla suojatun verkon sisällä, joten sen tietosuoja on kunnossa. Tätä sivustoa tullaan kuitenkin mahdollisesti käyttämään prototyypinä Mikamatic Oy:n tarjoamasta pääohjelmasta, jonka kääntämisestä mobiililaitteille sopivaksi on aina silloin tällöin keskusteltu. Tätä varten sivuston jatkokehityssuunnitelmiin kuuluu kirjautumissivun loppuun teko sekä tehokkuuden, tietoturvan ja ulkoasun parantaminen, sillä varsinaisessa asiakkaidemme tuotantotietokannassa on valtava määrä dataa. Sivusto on kuitenkin jo tällaisenaan käyttökelpoinen Mikamatic Oy:n sisäiseen käyttöön.

Annemari Mustonen

The development of mobile device compatible website MikamaticMobiili

Year	2020	Pages	30
------	------	-------	----

In this thesis, I focused on developing a mobile device compatible website MikamaticMobiili that will fetch information about people and companies stored in the Oracle database of Mikamatic Oy. The work began by examining Mikamatic Oy:s existing Yritysrekisteri - program that already contains these functionalities. The planning began with MarvelApp-application that I had planned on using. However, as the coding began the interface and general layout development of the website also gradually shifted to Visual Studio Code -program that was also used to do the coding. In the beginning I used Ionic Framework for the programming, but soon found out that there were big problems in connecting the site to the Oracle database using that framework without buying expensive external components. For this reason we stopped using Ionic Framework and began coding directly with HTML and Typescript programming languages.

The website itself was done using so-called Single Page Application style, also present in other Mikamatic Oy:s programs, where the main page always stays the same but its' content changes depending on the actions of the user. Within this one page I built tables for both people and companies inside a frame that shows basic information about them. I then created pages to show more detailed information about a single person or a company. This information can be accessed by selecting that person or company from the table on the main page and clicking the Katso-button. A login page was created, but at this point the user cannot give their login information as the website does that automatically.

The website only works within the network of Mikamatic Oy that has very good security through vpn connections, antivirus software and firewalls, so at this point there are no problems with information security. This website will possibly be used as a prototype version of Mikamatics' main program that our clients use as we have discussed translating that program for mobile devices. Because of that, the future development of the website done in this thesis will focus on completing the development of the login page as well as improving the efficiency, security and general design of the page as Mikamatics' main program contains much more data. The website is however usable for the purposes of Mikamatic Oy as it is.

Keywords: MikamaticMobiili, mobile device website, programming

Sisällys

1	Johdanto.....	6
1.1	Tavoitteet ja rajaus	6
1.2	Tietoturva ja eettisyys	7
1.3	Keskeiset käsitteet ja lyhenteet	8
2	Työn tietoperusta ja taustoitus	11
2.1	Valmiit komponentit	11
2.2	Versionhallinta.....	12
2.3	Ohjelman ja sen ohjelmoinnin yleiset periaatteet	12
2.4	Suunnittelu.....	14
3	Työn toteutus	14
3.1	Sivusto	15
3.2	Opinnäytetyöhön kehitetyt ja muokatut komponentit.....	21
4	Työn tulokset	22
4.1	Sivusto	22
4.2	Jatkokehitys	27
4.3	Arviointi	27
	Lähteet.....	28
	Kuviot	30

1 Johdanto

Opinnäytetyön nimi on ”Mobiilisivuston MikamaticMobiili kehitys”. Se toteutettiin toiminnallisena opinnäytetyönä. Projektin työelämäedustajana toimii espoolainen vuonna 1989 perustettu Mikamatic Oy, jonka päätoimiala on ohjelmistojen suunnittelu ja valmistus. Mikamatic Oy:n kehittämä MMC -tehdastietojärjestelmä on käytössä monilla suurilla elintarviketehtailla. Itse olen ollut kyseisessä yrityksessä töissä ohjelmoijana noin 7 vuotta.

Mikamatic Oy on pienestä koostaan huolimatta yritys, joka toimittaa ohjelmistoja suurille asiakkaille. Näistä asiakkaista tärkein on Valio Oy, jonka jokaisessa 12 toimipaikassa on käytössä yrityksen ohjelmisto. Mikamatic Oy:llä on myös ympärivuorokautinen päivystys vikatilanteiden varalta. Tämän vuoksi yrityksellä on erittäin laaja asiakasrekisteri, jota säilytetään Oracle Database 19c Standard Edition - tietokannan tauluissa. Rekisteriä käytetään yrityksen oman Yritysrekisteri - ohjelmiston kautta, joka hakee tiedot tietokannasta erilaisten hakutermien mukaan.

Olen kiinnostunut mobiiliohjelmoinnista ja olen opiskellut sitä nyt noin vuoden. Se on erittäin nopeasti kehittyvä ala, jolla on hyvät kasvunäkymät erityisesti globaaleilla markkinoilla, joten opinnäytetyön tekeminen tältä alalta on kannattavaa. Varsinaista aihevalintaa tehdessäni pohdin kahden eri mahdollisuuden välillä. Ensimmäinen olisi ollut enemmän omaan käyttöön tuleva koiriin liittyvä mobiilisovellus, jonka avulla koirien tärkeät tiedot olisivat kulkeneet kätevästi aina mukana. Samankaltaisia sovelluksia on kuitenkin jo tarjolla, joten päädyin jättämään tämän idean toistaiseksi toteuttamatta. Toiminnallisen opinnäytetyöni aiheeksi valikoitui siis suoraan työyhteisöni käyttöön tuleva rajattu käänös jo olemassa olevasta Yritysrekisteri-ohjelmasta. Aiheen valinta osui tähän opinnäytetyöhän myös siksi, että pääasiakkaamme Valio on osoittanut kiinnostusta kehittää Mikamatic Oy:n pääohjelmasta MMC:stä mobiiliversio jossain vaiheessa. Kyseinen ohjelmisto on ollut käytössä jo lähes 20 vuotta ja on erittäin monimutkainen, jatkuvasti kehityksessä oleva projekti. Tähän opinnäytetyöhön tuo projekti olisi kuitenkin ollut liian laaja ja monimutkainen. Vaikka tämän opinnäytetyön tuloksena syntyvä mobiilisivusto tuleekin vain oman yrityksemme sisäiseen käyttöön, siitä on tarkoitus tehdä tarpeeksi laadukas ollakseen mahdollisesti pohja tuolle laajemmalle, asiakkaille tarkoitettulle sovellukselle.

1.1 Tavoitteet ja rajaus

Yritysrekisteri-ohjelma sisältää monia eri toiminnallisuuksia, joista tähän opinnäytetyöhön tullaan sisällyttämään vain osa jotta siitä ei tulisi liian laaja. Rajaus on tarpeellinen myös ohjelman sisältämän suuren datamäärän takia.

Opinnäytetyön tavoite oli luoda mobiilisivusto Mikamatic Oy:n sisäisessä käytössä olevan Oracle -relaatiotietokannan henkilö- ja yritys tai toimipiste -tietoihin. Tietokannassa on n. kahden tuhannen henkilön ja noin tuhannen yrityksen sekä toimipisteen tiedot. Em. tietoja ylläpidetään ja käytetään päivittäin yrityksen normaalissa toiminnassa, mutta tämä on nykyisin mahdollista ainoastaan Windows -käyttöjärjestelmässä toimivalla Client-Server -tyyppisellä sovelluksella, Yritysrekisteri-ohjelmalla.

Tässä opinnäytetyössä kehitetyn mobiilisivuston kautta pystyy selaamaan sekä päivittämään tietokannassa olevia henkilö- ja yritystietoja. Sivuston kehitystä on tarkoitus jatkaa edelleen niin, että sillä pystyisi myös soittamaan ja lähettämään sähköpostia valitulle henkilölle tai yritykselle. Se tulee mahdollisesti olemaan myös esimerkki Mikamatic Oy:n asiakkaille, jos he jossain vaiheessa haluavat kehittää Mikamatic Oy:n MMC-ohjelmistosta mobiiliversion.

Tämän opinnäytetyön koodauksessa käytettiin paljon samoja tekniikoita ja yleisiä tapoja mitä Mikamatic Oy:n ohjelmissa yleensäkin on käytetty. Myös ulkoasu pyrittiin pitämään mahdollisimman samantyyllisenä kuin yrityksen muut ohjelmat, kuitenkin sitä suoraan kopioimatta. Tämä auttaa yhtenäisyyden saavuttamisessa, mutta on myös ohjelmoi-jille hyödyllistä. Kun ohjelmia tehdään tällä tavalla, muiden ohjelmoijien on helpompaa jatkaa toisen ohjelmoijan työtä. Myös alkuperäiselle ohjelmoijalle tämä on tärkeää, sillä harva pystyy helposti muistamaan esimerkiksi jonkin tietyn funktion tarkkaa tehtävää vuoden päästä. Kovin yksinkertaisia ohjelmapätkiä tehdessä tämä ei välttämättä ole aivan niin tärkeää, mutta heti kun ohjelmasta tulee hiemankin vaativampi se auttaa mm. virheiden etsimisessä. Kun ohjelmasta tulee todella pitkäikäinen ja valtavan määrän toiminnallisuksia omaava, on aivan erityisen tärkeää pitäytyä näissä ohjeissa.

Mikamatic Oy:n pääohjelma MMC on ollut kehityksessä jo noin 20 vuotta, joten sen kehityksessä on aikojen saatossa tullut niin paljon ominaisuuksia että niistä lukua pitäminen olisi käytännössä mahdotonta ilman erittäin tarkkaa pohjatyötä kunkin ohjelman osasen kehityksessä. Muutama vuosi sitten tehdyssä selvityksessä laskettiin, että kyseisessä ohjelmassa on noin 400 000 riviä toiminnallista koodia. Tästä opinnäytetyöstä on valmistuessaan tarkoitus tarjota Mikamatic Oy:n asiakkaille malli MMC:n mahdollisesta kääntämisestä mobiiliversioksi, joten yhtenäisyys alkuperäisen ohjelmiston kanssa on erityisen tärkeää.

Käytettävä Mikamatic Oy:n tietokanta on Oracle Database 19c Standard Edition 2, johon tarvittaessa ohjelmoidaan mobiilisivuston tarvitsemat lisätoiminnot Oracle PL/SQL - kielellä.

1.2 Tietoturva ja eettisyys

Sovelluksen tietoturva tulee suurimmaksi osaksi Mikamatic Oy:n laitteiden virustorjuntaohjelmistojen kautta. Testilaitteet, joihin sovellus on asennettu ovat kaikki

oman internet-verkkomme piirissä ja siten hyvin suojattuja, koska reitittimen palomuuuri estää ulkopuoliset yhteydet. Vielä tässä vaiheessa sivustoa ei tulla käyttämään oman suojatun internet-verkkomme ulkopuolella. Tietojen lataus tietokannasta vaatii käyttäjätunnuksen ja salasanan. Myös sovellukseen on ohjelmoitu erikseen kirjautuminen, mutta toistaiseksi se on suoraan kirjattuna koodiin eikä käyttäjä voi kirjautumistietoja muuttaa.

Tämän aiheen etiikka koskee hyvin pitkälti henkilötietojen käsittelyä. Sovelluksessa näkyvät ja siihen tallennetut puhelinnumerot ovat myös muuten kaikkien löydettävissä, mutta siihen ei tallenneta esimerkiksi salaisia numeroita. Sovelluksen tiedot tulevat hyvin rajatun käyttäjäkunnan tietoon, sillä meidän yrityksessämme työskentelee tällä hetkellä vain 3 henkilöä, kaikki perheemme jäseniä. Suurin osa sovelluksessa näkyvistä tiedoista on julkista tietoa esimerkiksi yritysten omista yhtiökertomuksista. Ainoa poikkeus tähän on osa sähköpostiosoitteista, mutta nämäkin tulevat meille tietoon asiakkaiden kautta ja heidän suostumuksellaan. Tarvittaessa heidän tietonsa voidaan tietokannasta poistaa. Tietoturva yrityksessämme on hyvä reitittimien asetusten, erillisten kertakäyttösalasanojen sekä tehokkaiden tietoturvaohjelmistojen ansiosta. Tämä pätee myös MikamaticMobiiliin.

1.3 Keskeiset käsitteet ja lyhenteet

Mobiilisovellus

Kännykälle tai tabletille tehty kohdennettu ohjelma, joka ei sellaisenaan toimi esimerkiksi suoraan tietokoneella.

Mobiilisivusto

Mobiilisivustolla tarkoitetaan yleisesti mobiililaitteita varten kehitettyä internet-sivustoa. Tällaista sivustoa voi kyllä käyttää myös muilla laitteilla, mutta sen ulkoasu ja suorituskyky on optimoitu mobiililaitteita varten.

Mobiiliohjelmointi

Mobiiliohjelmoinnilla tarkoitetaan kännykälle tai tabletille tehtävää erillisellä ohjelmointikielellä toteutettavaa ohjelmointia.

MarvelApp

Mobiilisovellusten ja niille suunnattujen sivustojen suunnitteluun tarkoitettu ohjelma. (Marvel 2020)

Relaatiotietokanta

Relaatiotietokanta on tietokantamalli, jonka perustana ovat niin kutsutut taulut, joihin varastoidaan tietoa ja joiden välille muodostetaan yhteyksiä niin sanottujen viiteavainten avulla. Viiteavaimet yhdistävät taulujen tiedot toisiinsa. Relatiotietokannassa viiteeheyksillä varmistetaan tietoyhteyksien säilymistä ja kontrolloidaan mitä tapahtuu eri tauluissa säilytetyille tiedolle siinä tapauksessa, että sitä muutetaan yhdessä taulussa. Taulujen välisiä suhteita voi usein kuvata äiti-lapsi termillä siten, että jollain tietyllä äititaululla voi olla useita lapsitauluja, mutta lapsitaululla voi olla vain yksi äititaulu. Relatiotietokannassa sallitaan myös moni-moneen suhteet taulujen välillä, sekä linkkitaulujen käyttö. (Sarja 2006)

Web-palvelin

Jotta tiedot saadaan liikkumaan mobiilisivuston ja relaatiotietokannan välillä, on perustettava web-palvelin. Web-palvelin on laite, ohjelmisto tai niiden yhdistelmä, joka tarjoaa www-sivuja. Sitä voidaan käyttää myös ihmisiltä piilossa olevan tiedon kuljettamiseen eri alustojen välillä. (What is a web server? 2020)

Client-Server -sovellus

Client-Server -sovellus tarkoittaa ohjelmaa, joka hakee tietonsa palvelimen tietokannasta sen sijaan, että ne olisi talletettu itse ohjelmaan tai sen käyttäjän laitteelle. (Client-Server Overview 2020)

Ionic Framework

Ionic Framework on pääasiassa mobiilisovellusten ja -sivustojen kehitykseen tehty sovelluskehitin, jota tullaan tässä projektissa käyttämään. (Ionic Framework 2020)

SQL

SQL on lyhenne sanoista Structured Query Language, ja sitä käytetään SQL-kyselyissä joilla haetaan dataa relaatiotietokannasta. (SQL - Overview 2020, 1)

PL/SQL

PL/SQL on lyhenne sanoista Programming Language / Structured Query Language. Sen avulla datan lisääminen, muuttaminen ja hakeminen SQL - tietokannasta on tehokkaampaa. (PL/SQL - Overview 2020, 1)

API

API on lyhenne englanninkielisistä sanoista Application Programming Interface. API:n avulla sovellus keskustelee tietokannan kanssa.

HTML

HTML eli hypertext markup language on ohjelmointikieli, jota käytetään lähinnä verkkosivustojen tekemiseen. (HTML: HyperText Markup Language 2019)

CSS

CSS on lyhenne englanninkielisistä sanoista cascading style sheet, ja sen avulla kehitetään sivuston tai sovelluksen ulkonäköä. (CSS Tutorial 2020)

Javascript

Javascript on ohjelmointikieli, jota käytetään yhdessä HTML:n ja CSS:n kanssa verkkosivustojen teossa. Se on kehitetty Java - ohjelmointikielestä. (PluralSight 2020)

Typescript

Typescript on Javascripti-ohjelmointikielen kehittyneempi versio. (Typescript 2020)

Delphi

Delphi on eräs Pascal - ohjelmointikielestä kehitetty versio, jota saman niminen ohjelmisto käyttää (Embarcadero 2016). Mikamatic Oy:ssä tätä kieltä ja ohjelmistoa käytetään paljon.

Oracle Database 19c Standard Edition 2

Oracle Database 19c Standard Edition 2 on Oraclen kehittämä, PL/SQL:ään pohjautuva relaatiotietokanta (Oracle 2020), jota Mikamatic Oy käyttää kaikessa päivittäisessä toiminnassaan ja johon myös tämän opinnäytetyön käännettävä Yritysrekisteri - ohjelma pohjautuu.

Single Page Application (SPA)

Single Page Application tarkoittaa sen tyyppistä internetsivustoa, jossa sivu itsessään pysyy samana mutta sen sisältö muuttuu käyttäjän navigoidessa sivustolla. (SPA (Single-page application) 2020)

Visual Studio Code

Visual Studio Code on Microsoft Oy:n kehittämä ohjelmisto, joka on tarkoitettu ohjelmien ja sivustojen kehittämiseen eri ohjelmointikielillä. (Microsoft 2020)

TortoiseHG

TortoiseHG on graafinen versionhallintaohjelma, joka tukee Mercurial - versionhallintajärjestelmää. Versionhallinnan kautta voi hallita ja säilyttää ohjelmistokehityksen aikaisia erilaisia versioita kehitettävästä ohjelmasta tai muusta vastaavasta projektista. (Borho, 2020)

Mercurial

Mercurial on ilmainen, helppokäyttöinen versionhallintajärjestelmä (Work easier work faster, 2020)

BitBucket

Bitbucket on pilvipalvelu, johon on mahdollista säilöä dataa ja jakaa sitä muiden kehittäjien kanssa. (A brief overview of Bitbucket 2020)

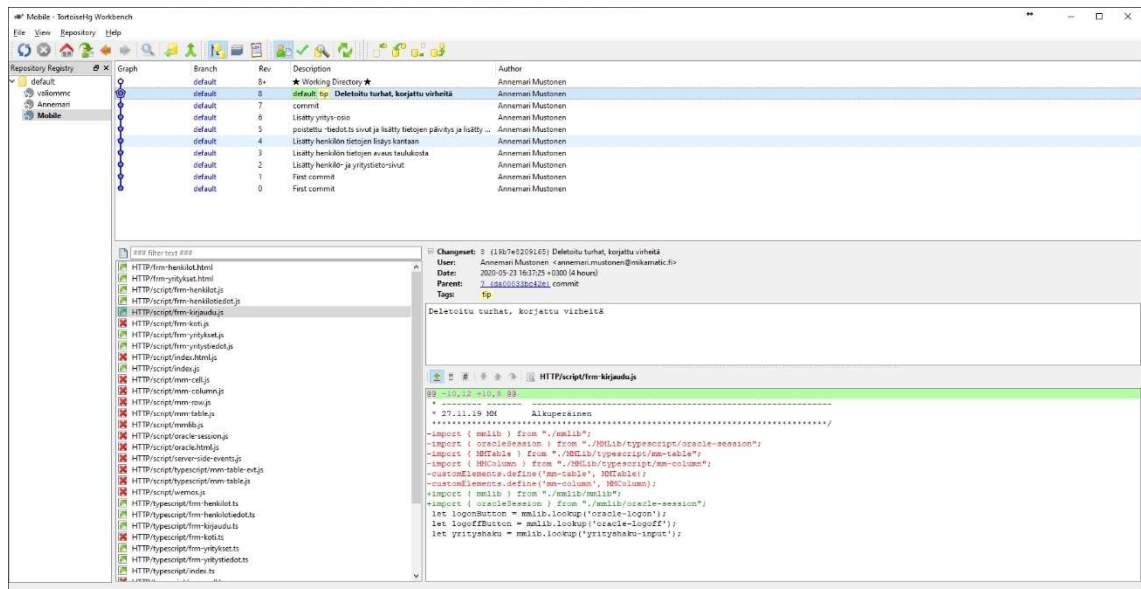
2 Työn tietoperusta ja taustoitus

2.1 Valmiit komponentit

Osa opinnäytetyössä käytetyistä ohjelmistokomponenteista on Mikamatic Oy:n valmiiksi aiemmin kehittämiä. Niihin kuuluu myös opinnäytetyössä käytetyn Web-palvelimen perusta ja sen API-rajapinta. Sen lisäksi valmis komponentti on MMLib.ts. Se on typescript-kirjasto, jossa on kehittyneempiä yleistoimintoja. Sen avulla sivusto hakee älykkäästi index.html - sivun elementtejä niin, että se tarkistaa onko kyseinen elementti itse pääsivulla vai sen sisältämässä framessa. MMLib.ts on alun perin suunniteltu automaatioelementtien tueksi, sisältäen mm. ääniefektejä, joita ei tässä opinnäytetyössä kuitenkaan käytetty. Tätä ja vastaavia kirjastoja käytetään myös yrityksen muissa ohjelmissa.

2.2 Versionhallinta

Ohjelmointityötä tehdessä on erittäin tärkeää varmistua siitä, että mahdollisten suurien virheiden varalta voi aina helposti palata aiempaan, toimivaan ohjelmaversioon. Tätä varten on kehitetty erilaisia versionhallintaohjelmistoja. Tähän tarkoitukseen Mikamatic Oy käytti opinnäytetyön tekoa aloitettaessa Mercurial - versionhallintajärjestelmää, BitBucket - palvelua varsinaiseen datan säilytykseen sekä TortoiseHG - käyttöliittymää (Kuvio 1) sen graafiseen hallintaan. Myös tämä opinnäytetyö käyttää noita järjestelmiä, mutta Mikamatic Oy joutui siirtymään käyttämään Git - versionhallintajärjestelmää sekä sen GitHub - palvelua BitBucketin lopetettua Mercurialin tukemisen. Mercurial toimii kuitenkin edelleen ilman mahdollisuutta tallentaa tietoja pilveen, joten toistaiseksi tämän opinnäytetyön versionhallinta on edelleen siellä. Se on myös varmuuden vuoksi kopioitu toiselle tietokoneelle siltä varalta, että ensisijaiselle ohjelmointitietokoneelle tapahtuu jotain.



Kuvio 1: TortoiseHg-käyttöliittymä

2.3 Ohjelman ja sen ohjelmoinnin yleiset periaatteet

Pyrin kaikessa koodauksessa käyttämään selkeää kappale- ja funktiojakoa sekä tekemään koodista mahdollisimman luettavaa. Jokaiselle toiminnallisuudelle pyritään antamaan yksilökohtainen nimi, sen toiminnasta kirjoitetaan koodiin lyhyt kommentti, ja tiedostojen ylälaitaan kommentoidaan perustiedot tiedoston toiminnasta sekä joissain tapauksissa lyhyesti myös aikojen kuluessa tehdyt muutokset. Funktioiden ja muiden osioiden nimeämisessä pyritään olemaan mahdollisimman johdonmukainen, jotta niitä kutsuvaa koodia on helppo kirjoittaa, eivätkä funktiot mene sekaisin. Ohjelman alalaidassa on pieni viestikenttä, joka kertoo käyttäjälle hieman lisätietoa hänen juuri suorittamastaan toiminnasta tai esimerkiksi vastaan tulleesta virheestä. Myös tiedostojen nimet tukevat tätä toimintamallia, frm-etuliite

kertoo että kyseessä on form - tyylinen ohjelmakoodia sisältävä tiedosto. Sivuston koodiosa on pyritty jakamaan mahdollisimman hyvin ulkonäkö - osioon sekä funktionaaliseen osioon selkeyden vuoksi. Käytännössä tämä tarkoittaa sitä, että HTML - tiedostossa on koodattuna sivuston yleinen ulkoasu sekä kutsut ja TS - tiedostossa tapahtuu varsinainen tiedonhaku sekä muut funktiot. Tätä käytetään myös yrityksen muissa ohjelmissa.

Opinnäytetyössä käytetään yrityksen muista ohjelmista tuttua Single Page Application - tyyliä, eli käyttäjän klikkaamalla sivuston eri osia ja nappuloita sivu pysyy samana, mutta sen sisältö vaihtuu. Koodiin on myös jätetty kommentteihin joitain kokeiluja sekä niin sanottuja hälytyksiä, joiden avulla helpotetaan tulevaisuudessa ongelmien etsintää sekä koodin päivytystä (Kuvio 2). Kommentoitu koodi ei normaalisti käynnisty ollenkaan, vaan kääntäjä ohittaa sen ellei ohjelmoija poista koodinpätkästä kommenttimerkkejä.

```

typescript > TS frm-henkilots > window.onload
WHERE sukunimi LIKE '${henkilohaku.value}' OR
  toimipiste IN
  ( SELECT toimipiste_id
    FROM toimipiste
    WHERE yritys nimi LIKE '${henkilohaku.value}' )
ORDER BY sukunimi, etunimi
);

mmTable.pageNumber!.value = '0';
mmTable.load({forward:true, advance_page:true});
});

/*
mmlib.lookup('execute-sql').addEventListener('click', () => {
  let sqlSelectElement: HTMLTextAreaElement = mmlib.lookup('sql-select') as HTMLTextAreaElement;

  if (sqlSelectElement) {
    oracleSession.select(sqlSelectElement.value, function (response: string){
      mmlib.notify('sql-result', response);
      mmlib.notify('info', 'Rivit ladattiin', false, false, true);
    });
  }
});

mmlib.lookup('clear-sql').addEventListener('click', () => {
  mmlib.notify('sql-result', '...');
});
*/

henkilohaku.addEventListener('keydown', (e: KeyboardEvent) => {
  if (e.keyCode == 13) {
    e.preventDefault();
    mmlib.notify('info', 'Haetaan yrityksen ${henkilohaku.value} henkilöitä ...');
    mmTable.load({forward:true, advance_page:false});
  }
});

//ladataan henkilöt
let mmTable:MMTable = mmlib.lookup('henkilot-table') as MMTable;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

> Executing task: tsc -p d:\Mobile\HTTP\tsconfig.json <
Terminal will be reused by tasks, press any key to close it.

```

Kuvio 2: Kommentoitua koodia

2.4 Suunnittelu

Opinnäytetyö oli tarkoitus suunnitella MarvelApp - nimisellä internetissä olevalla työkalulla. Sen käyttö on minulle tuttua jo aiemmista projekteista. Kuitenkin kuten joskus näissä projekteissa käy, myös ulkonäön suunnittelu siirtyi pikkuhiljaa suoraan koodausohjelmiston sisään. Se on tietyllä tavalla helpompaa, mutta toisaalta näin tehtäessä osa design - ideoista jää aina toteutumatta kehitysohjelmiston kankeuden vuoksi. Sivustosta haluttiin kuitenkin Mikamatic Oy:n jo olemassa olevan ohjelman Yritysrekisterin kaltainen, joten jo se rajoitti suunnittelua jonkin verran.

Käytännössä sivuston ulkoasu siis muokkautui yhtä aikaa ohjelman kehityksen kanssa. Tämä tapahtui vesiputousmallia noudattaen, ensin jokin osio suunniteltiin Visual Studio Codessa, se toteutettiin ja sitten testattiin, jonka jälkeen palattiin tarvittaessa takaisin suunnitteluvaiheeseen ja niin edelleen. Tällä tavalla kokeiltiin erilaisia värvaihtoehtoja sekä taulukon rivien määriä, leveyksia ja headerin ja footerin kokoa. Se hyöty tämän tyylisestä suunnittelusta oli, että heti kävi selväksi toimisiko jokin asia talukossa jossa on dataa enemmän. Tämä olikin asia, jota jouduttiin moneen kertaan kokeilemaan ja korjaamaan.

3 Työn toteutus

Sivuston ja tietokannan väliin konfiguroitiin web-palvelin, jonka välityksellä mobiilisivusto kommunikoi tietokannan kanssa. Tähän Mikamatic Oy:llä on Delphi -sovelluskehittimellä toteutettuja valmiita malleja, mutta tätä opinnäytetyötä varten niistä muokattiin oma web-palvelin.

Opinnäytetyön toteutus tapahtui Visual Studio Code - ohjelmistolla. Toteutus aloitettiin käymällä ensin tarkasti läpi Mikamatic Oy:n Yritysrekisteri-ohjelma, joka toimii sivuston mallina. Yritysrekisteri - ohjelmisto on laaja-alainen ohjelma, joka sisältää muitakin toiminnallisuuksia kuin tähän opinnäytetyöhän tarvittavat osat. Toteutukseen sisältyi myös tärkeänä osana jo lähes valmiiksi rakennettu Web-palvelin sekä muita Mikamatic Oy:n kehittämiä komponentteja. Toteutukseen jäi itse tämän sivuston suunnittelu, teko ja niiden valmiiden komponenttien integrointi sopivaksi osaksi sitä.

Opinnäytetyön alkupuolella koitettiin pysyä alkuperäisessä suunnitelmassa, joka oli käyttää Ionic - mobiilisovelluskehitystyökalua, siihen liittyvää Apache Cordova sekä Angular - ohjelmointikieltä. Suhteellisen pian kävi kuitenkin ilmi, ettei tiedonhaku Oraclen tietokannasta tulisi onnistumaan ilman erittäin vaativaa ja kallista koodaus- ja kehitystyötä. Oraclen tietokanta on sen verran vähän käytetty yritysmaailman ulkopuolella, ettei siihen soveltuvia valmiita komponentteja jotka olisivat olleet hintaluokaltaan sopivia oikein

löytynyt. Tästä syystä pitkän pohdinnan jälkeen päätettiin siirtyä käyttämään Mikamatic Oy:n omia komponentteja ja web - palvelinta.

Sovellus hakee tiedot tietokannasta tapahtuu API-rajapinnan kautta. Tietokantayhteys on ohjelmoitu oracle-session.ts - tiedostoon. Myös tämä tiedosto on osittain valmis komponentti, jota on tätä opinnäytetyötä varten muokattu. Tietokantaan kytkeydytään opinnäytetyössä tehdyllä frm - kirjaudu - toiminnolla.

3.1 Sivusto

Opinnäytetyön koodausvaihe aloitettiin luomalla index.html ja index.ts - tiedostot. Index-sivu on ohjelman pääsivu, jonka sisään kaikki muut komponentit latautuvat. Kuten on tyypillistä yrityksemme muillekin ohjelmille ja yleiselle koodausperiaatteelle, index.ts sivu itsessään sisältää vain vähän koodia sillä sen pääfunktio on kutsua muita sivuja. Se sisältää menun, valmiin iFrame-komponentin johon tiedot latautuvat sekä alalaidan viestikentän, joka kertoo hieman lisätietoa asioista.

Index.ts- tiedostoon on koodattu menun lataus ja navigointi window.onload - funktioon, jonka avulla iFrame - komponenttiin ladataan menun kolme eri sivua. Nämä sivut ovat Henkilöt, Yritykset ja Kirjaudu. Siihen kuuluu olennaisena osana myös Mikamaticin oman mmlib - luokan lataus.

Suurin työ opinnäytetyössä tapahtui henkilölista-osiota kehittäessä, sillä siitä tuli malli myös yrityslistalle. Tätä varten luotiin kaksi tiedostoa, frm-henkilot.html ja frm-henkilot.ts. HTML-tiedostoon tuli sivuston ulkonäkö, painikkeet ja niiden perustoiminnallisuudet. Frm-henkilot.ts -tiedostoon koodattiin yksityiskohtaisemmat toiminnallisuudet, joita HTML-tiedosto kutsui. Sivun haki muotoaan kauan ja suurin osa koodauskokeiluista, testeistä ja myös virheistä tapahtui tätä sivua tehdessä.

Aluksi tälle henkilölista-sivulle kovakoodattiin sekä kirjautuminen että alkuperäinen SQL select - lause, joka haki alustavat tiedot tietokannasta (Kuvio 3). Kovakoodaus tarkoittaa sitä, että kyselyn tarkat parametrit kirjoitetaan suoraan koodiin ilman, että käyttäjä voi niihin vaikuttaa. Tämä ei ole hyvän ohjelmointitavan mukaista, mutta monesti aluksi on nopeinta kovakoodata parametrit suoraan koodiin, jotta sivuston perusteet saadaan tehtyä. Tämän jälkeen aletaan siirtää ja siistiä koodia.

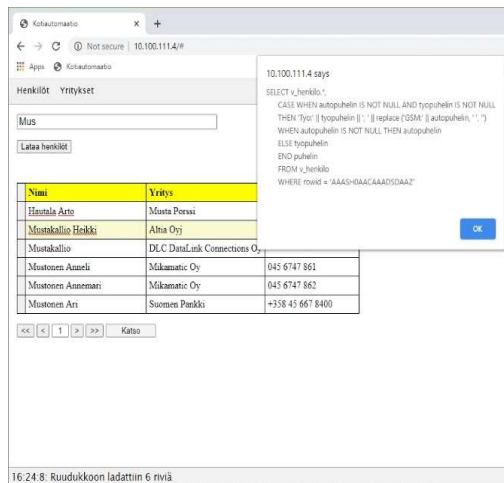
```
<body>
</div>
<!-- MM-TABLE -->
<div class="row">
  <div class="col-12">
    <input type="text" id="yrityshaku-input"
      placeholder="Yrityshaku" size="40" value="Valio" style="font-size: medium;"
    >
  <p/>
  <button id="load-yritykset">Lataa yritykset</button>
</div>
</div>
```

Kuvio 3: Kovakoodattu hakueto

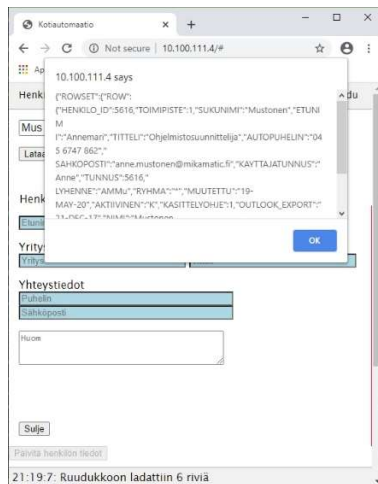
HTML - sivun tärkein komponentti on taulukko, johon kaikki haetut tiedot tulevat. Aluksi taulukko oli hyvin yksinkertainen, siinä näkyi pelkästään henkilöiden nimiä eikä siinä ollut edes hakumahdollisuutta, vaan esimerkkihaku oli kovakoodattuna .ts - tiedostoon. Tällä varmistettiin tietokantalinkin toimivuus ja pystyttiin menemään ohjelmoinnissa eteenpäin.

Tällä frm-henkilot.html - sivulla tapahtui suurin osa sivuston ulkonäön suunnittelusta ja teosta. Sivuston tekoa jatkettiin eteenpäin vasta, kun olin tyytyväinen tämän osion ulkonäköön. Mobiililaitteiden rajoitetun näyttökapasiteetin takia ulkonäkö oli pakko jättää suhteellisen yksinkertaiseksi.

Kunhan HTML-sivun ulkoasu ja perusfunktiot saatiin kuntoon, siirrettiin varsinainen koodaus ja tapahtumat frm-henkilot.ts - tiedoston puolelle. Aluksi myös siellä oli vain yksinkertainen select - lause, joka haki tietokannasta henkilöiden nimet. Ensin tietoja ei edes ladattu itse taulukkoon, vaan testattiin tietokantahaun toimivuutta ainoastaan tulostamalla ne alert - funktioon, joka avaa popup - ikkunan jossa tiedot näkyivät (Kuvio 4 ja 5). Tämä nopeuttaa kehitykseen käytettävää aikaa ja vähentää sivuston välisen navigoinnin tarvetta kehitys- ja testausvaiheessa. Osittaisen valmiiden komponenttien käytön vuoksi tässä tiedostossa tuli ajoittain ongelmia kansiorakenteen kanssa, ne kuitenkin onnistuttiin ratkomaan yksinkertaisesti oikean tiedostopolun käytön avulla.



Kuvio 4: Esimerkki PopUp - ikkunasta



Kuvio 5: Toinen esimerkki PopUp - ikkunasta

Kun perushaku saatiin yllä mainitulla tavalla toimimaan, alettiin sql - kyselyä laajentamaan haluttuihin tietoihin. Tietokannasta haluttiin hakea henkilön nimi, työpaikka ja puhelinnumero ja tulostaa ne lyhyeen listaan. Listaan otettiin näkyviin vain 6:n henkilön nimet, jotta niitä olisi mahdollisimman helppo lukea mobiililaitteella. Taulukko itsessään on valmis komponentti MM - Table, jonka perustoiminnot ovat omassa tiedostossaan. Sitäkin jouduttiin jonkin verran muuttamaan, jotta taulukko sopisi hyvin juuri tähän sivustoon.

Vaikeinta tässä vaiheessa tämän sivun kehityksessä oli saada kunnollinen sql - lause siihen integroitua niin, että se haki kaikki tarvittavat tiedot. Ongelmalliseksi tämän teki se, että monilla henkilöillä on tietokannassamme useampi puhelinnumero. Niitä oli pakko karsia ja yhdistellä jonkin verran, jotta ne saatiin näkymään mobiililaitteellakin järkevästi. SQL - lauseessa tämä tehtiin case - lauseen muodossa (Kuvio 8). Case - lauseella voi Oracle PL/SQL - kielessä yhdistellä tietoja ja laittaa niille ehtoja. Koodissa nämä puhelinnumerot erotettiin

Työ - ja GSM - numeroiksi. Tämän sivun ohjelmoinnissa käytettiin usein hyväksi popup -ikkunoita, sillä niillä on helppo testata sql - lauseiden toimivuutta.

```

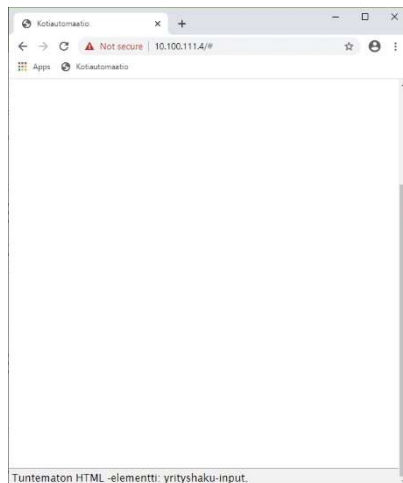
typescript > TS frm-henkilots > buildSQL
19 let henkilöId: string;
20 let henkilohakuInput: HTMLInputElement;
21 let etunimiInput: MMInput;
22 let sukunimiInput: MMInput;
23 let yritysInput: MMInput;
24 let titteliInput: MMInput;
25 let puhelinInput: MMInput;
26 let sahkopostiInput: MMInput;
27 let huomText: HTMLTextAreaElement;
28 let loadHenkilotButton: HTMLButtonElement;
29 let henkilotRow: HTMLTable;
30 let henkilotTable: MMTable;
31 let suljeHenkiloButton: HTMLButtonElement;
32 let paivitaHenkiloButton: HTMLButtonElement;
33
34
35 function buildSQL(): string {
36
37     return `SELECT rowid, nimi, yritysnimi,
38     CASE WHEN autopuhelin IS NOT NULL AND tyopuhelin IS NOT NULL
39     THEN 'Tyo:' || tyopuhelin || ', ' || replace('GSM:' || autopuhelin, ' ', '')
40     WHEN autopuhelin IS NOT NULL THEN autopuhelin
41     ELSE tyopuhelin
42     END puhelin
43 FROM v_henkilo
44 WHERE sukunimi LIKE '${henkilohakuInput.value}%' OR
45 toimipiste IN
46 ( SELECT toimipiste_id
47 FROM toimipiste
48 WHERE yritys nimi LIKE '${henkilohakuInput.value}%' )
49 ORDER BY sukunimi, etunimi
50 `;
51
52
53 window.onload = () => {
54     henkilohakuInput = mmlib.lookup('henkilohaku-input') as HTMLInputElement;
55     loadHenkilotButton = mmlib.lookup('load-henkilot') as HTMLButtonElement;
56     henkilotRow = mmlib.lookup('henkilot-row') as HTMLTable;
57     henkilotTable = mmlib.lookup('henkilot-table') as MMTable;
58     henkilohakuInput = mmlib.lookup('henkilohaku-input') as HTMLInputElement;
59     suljeHenkiloButton = mmlib.lookup('sulje-henkilo-button') as HTMLButtonElement;
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Kuvio 6: Case - lause esimerkki

Aluksi oli tarkoitus käyttää vain yhtä taulukkoa sekä henkilöiden että yritysten hakuun. Selkeyden vuoksi päätin kuitenkin erottaa taulukot omille sivuilleen, ja tämä ratkaisu vaikuttaa toimivalta. Yrityslistaan pystyin suurelta osin kopioimaan henkilölistan koodia, sitä ei juuri tarvinnut muuttaa muuten kuin nimien ja hakuheitojen osalta. Yrityslistassa näytetään vain yrityksen nimi ja puhelinnumero. Koska yrityksillä ei tyypillisesti ole kuin yksi puhelinnumero, tällä sivulla ei ollut tarvetta käyttää sql - selectissä case - lausetta puhelinnumeroita yhdistelemässä.

Pieni virhe koodissa hävitti koko yrityslista - taulukon, sen script - kutsu oli jäänyt pois (Kuvio 9). Kun se palautettiin, taulukko toimi normaalisti (Kuvio 10 ja 11).



Kuvio 7: Sivu koodivirheen jälkeen

```

> _history
> .vscode
> script
  typescript
    mmlib
      TS mm-cell.ts
      TS mm-column.ts
      TS mm-input.ts
      TS mm-row.ts
      TS mm-table.ts
      TS mmlib.ts
13 |
14 | <!doctype html>
15 | <html lang="en">
16 | <head>
17 |   <link rel="stylesheet" href="/style.css">
18 |   <link rel="stylesheet" href="/mm-table.css">
19 |
20 |   <script type="module" src="/script/frm-yritykset.js"></script>
21 |   <script type="module" src="/script/mmlib/mm-input.js"></script>
22 | </head>
23 |
24 | <body>
  
```

Kuvio 8: Kohta, jossa virhe oli

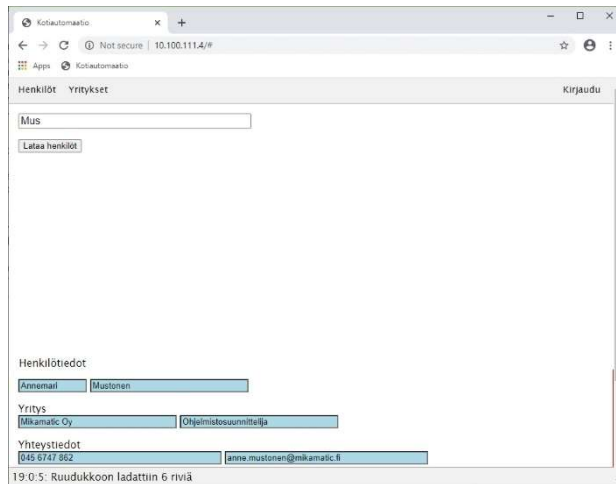
```

> _history
> .vscode
> script
  typescript
    mmlib
      TS mm-cell.ts
      TS mm-column.ts
      TS mm-input.ts
      TS mm-row.ts
      TS mm-table.ts
      TS mmlib.ts
13 |
14 | <!doctype html>
15 | <html lang="en">
16 | <head>
17 |   <link rel="stylesheet" href="/style.css">
18 |   <link rel="stylesheet" href="/mm-table.css">
19 |
20 |   <script type="module" src="/script/frm-yritykset.js"></script>
21 |   <script type="module" src="/script/mmlib/mm-input.js"></script>
22 |   <script type="module" src="/script/mmlib/mm-table.js"></script>
23 | </head>
24 |
  
```

Kuvio 9: Virhe korjattu

Yksi tämän opinnäytetyön tärkeimpiä toiminnallisuuksia on henkilöiden ja yritysten tietojen päivitys tietokantaan. Opinnäytetyöhön ei vielä ehditty tehdä uuden henkilön tai yrityksen lisäystä tietokantaan, mutta tietojen päivitys saatiin toimimaan. Aluksi tehtiin erikseen sekä HTML - että typescript - sivut kullekin toiminnolle, mutta koska tiedostojen lukumäärä koitettiin pitää mahdollisimman pienenä ja sivustorakenne yksinkertaisena, lopulta kaikki toiminnallisuus siirrettiin vastaavan listasivun typescript - tiedostoihin. HTML - sivut pidettiin erillisinä ja nimettiin opinnäytetyön teemaa vastaavasti frm-henkilötiedot.html ja frm-yritystiedot.html. Yksittäisten henkilöiden ja yritysten yksityiskohtaisiin tietoihin pääsee klikkaamalla taulukossa vastaavaa riviä ja sen jälkeen klikkaamalla Katso - nappia sivun alareunassa.

Tässä osassa ongelmaksi muodostui taulukon näyttäminen ja piilottaminen. Ne sotkivat koko sivun näyttämistä jonkin aikaa. Tämä ongelma ratkaistiin piilottamalla erikseen taulukko ja sitten yksitellen sen rivit. Se ei kuitenkaan aivan riittänyt, sillä jostain syystä taulukko jäi vielä kummittelemaan taustalle. Avautuvan sivun yläreunaan jäi valtava valkoinen alue, jonka korjaamiseen vaadittiin taulukon rivien korkeuden muuttaminen väliaikaisesti 0:ksi (Kuviot 12 ja 13). Sen jälkeen myös yksityiskohtaisten tietojen sivu avautui kuten pitääkin.



Kuvio 10: Valkoinen alue

```
//näytetään valitun henkilön tiedot Katso - nappia painettaessa ja piilotetaan taulukko
henkilotTable!.viewButton!.addEventListener('click', (e: MouseEvent) => {
  henkilotRow.style.visibility = "hidden";
  henkilotRow.style.height = "0";
  henkilotTable.style.visibility = "hidden";
  henkilöRow.style.visibility = "visible";
});
```

Kuvio 11: Korostetun rivin lisäys poisti ongelman

Yksittäisten henkilöiden tiedoissa haluttiin näyttää mahdollisimman monia asioita kuitenkin täyttämättä liikaa mobiililaitteen näyttöä. Niinpä näytettäviin henkilötietoihin valittiin näkymään etunimi, sukunimi, työpaikka, titteli, maksimissaan kaksi puhelinnumeroa, sähköposti sekä Mikamatic Oy:lle tärkeä huom - kenttä, johon voi lisätä vapaamuotoista kirjoitusta.

Yrityksen tietoihin valittiin vain nimi, puhelin, sähköposti sekä huom - kenttä. Tilan säästämisen vuoksi kaikkia kenttiä ei otsikoitu, mutta jos ne ovat tyhjiä, niissä näkyy placeholder - teksti joka kertoo mitä kenttään on tarkoitus kirjoittaa.

Tietojen päivitys Oracle - tietokantaan tapahtuu update - lauseella. Typescript - tiedosto havaitsee, kun käyttäjä tekee muutoksia input - kenttiin ja poimii ne sieltä, jonka jälkeen Päivitä henkilön tai yrityksen tiedot - nappi aktivoituu. Nappia painamalla uudet tiedot

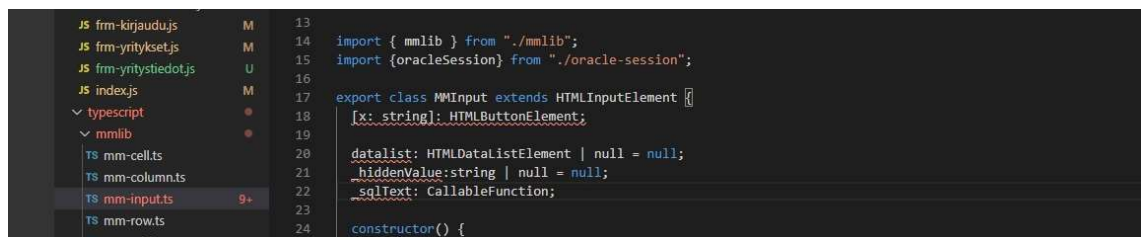
tallentuvat tietokantaan ja ovat heti näkyvissä sekä haettavissa. Sulje - napin käyttö peruuttaa muutokset jos sitä painetaan ennen Päivitä -nappia.

3.2 Opinnäytetyöhön kehitetyt ja muokatut komponentit

MMLib - luokka on kaikille Mikamatic Oy:n ohjelmille yhteinen nimitys tiedostosta, joka sisältää yleisiä funktioita ja muita pieniä toiminnallisuuksia. Sen sisältö vaihtelee ohjelmistokohtaisesti. Tässä opinnäytetyössä se on Javascript - apuohjelmakirjasto. Sen pääasiallinen toiminta liittyy lookup - funktioon, joka etsii HTML - elementtejä sivustolta. Tämä toiminnallisuus takaa sen, että iFrame - komponentti hakee aina halutut tiedot ja näyttää ne oikeassa paikassa.

MM - Table - komponentti on taulukkokomponentti, jota käytetään opinnäytetyön Henkilö- ja yrityslistoissa. Siihen on liitetty myös nappi, joilla voi kelata listan sivuja. Taulukon ulkoasuun tehtiin tätä opinnäytetyötä varten joitain muutoksia. Nämä muutokset tehtiin mm-table.css - tiedostossa. Niitä ovat otsikon ja aktiivisen rivin taustaväriin muuttaminen keltaiseksi, taulukon taustaväriin säätö, footerin koon säätäminen pienemmäksi sekä marginaalien pienennys. MM - Table - komponentin valmiita alikomponentteja, joihin ei tehty suuria muutoksia ovat MM - Column, MM - Row ja MM - Cell.

MM-Input typescript - tiedosto, joka on pääosin valmiskomponentti, aiheutti jonkin verran päänvaivaa. Sen tarkoituksena on helpottaa erilaisten input - elementtien tekoa, ja tämän komponentin pienen koodausvirheen takia HTMLButtonElement, jota muu sivusto käyttää, lakkasi toimimasta oikein (Kuvio 14). Tämän takia ohjelman kääntäminen ei enää onnistunut, vaan se pysähtyi useaan virheeseen, jotka kaikki liittyivät lopulta samaan asiaan. Virhe paikantui lopulta Visual Studio Code - ohjelmiston tarjoaman pikaratkaisun syyksi, sillä se oli generoinut mm-input - tiedostoon oman [x: string]: HTMLButtonElement; koodinpätkän toista virhettä korjatessa. Tästä näkee, että vaikka joskus kehitysohjelmistojen tarjoamat pikakorjaukset ja -ratkaisut ovat hyödyllisiä ja nopeuttavat toimintaa, niiden käytössä täytyy olla erittäin tarkkana. Aina ne eivät välttämättä sovi oman koodin kanssa yhteen.



```

13
14 import { mmlib } from "../mmlib";
15 import { oracleSession } from "../oracle-session";
16
17 export class MMInput extends HTMLInputElement {
18     [x: string]: HTMLButtonElement;
19
20     dataList: HTMLDataListElement | null = null;
21     hiddenValue: string | null = null;
22     sqlText: CallableFunction;
23
24     constructor() {

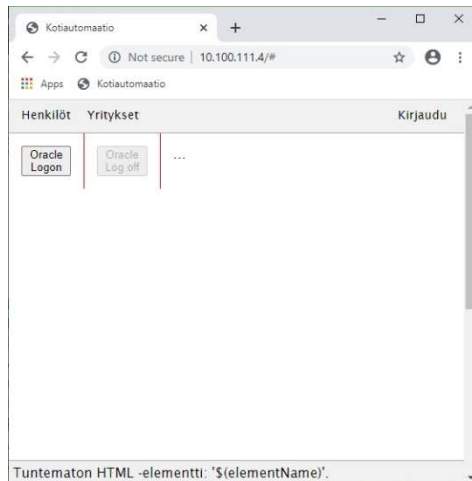
```

Kuvio 12: HTMLInput - virhe

4 Työn tulokset

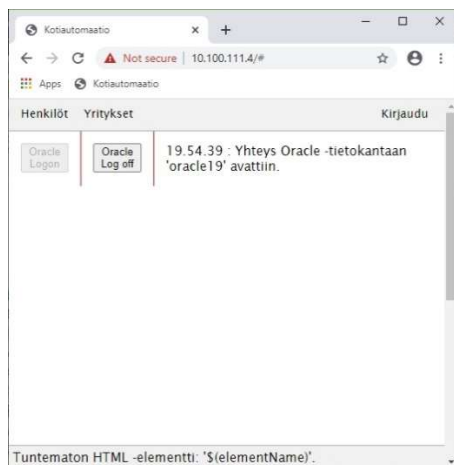
4.1 Sivusto

Tällä hetkellä ilman sisäänkirjautumista kaikki sivut aukeavat, mutta niihin ei voi hakea mitään dataa. Kirjautu - sivulta löytyy Oracle Logon ja Oracle Log off - napit, joiden avulla kirjautuminen sisään ja ulos tietokannasta hoidetaan (Kuvio 15).

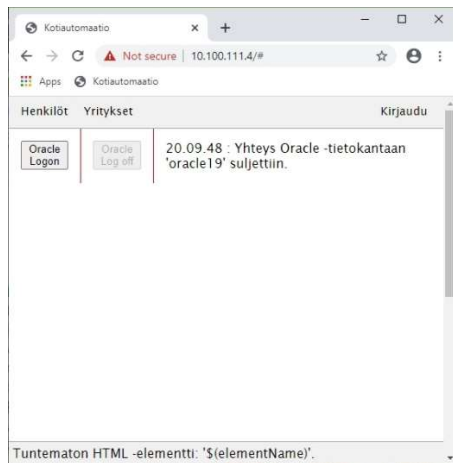


Kuvio 13: Kirjautu-sivu

Kun Oracle Logon - nappia painaa, käyttäjä kirjataan sisään sivustoon koodatulla käyttäjätunnuksella ja ilmoittaa siitä käyttäjälle. Log off - nappi kirjaa käyttäjän ulos (Kuvio 16 ja 17).

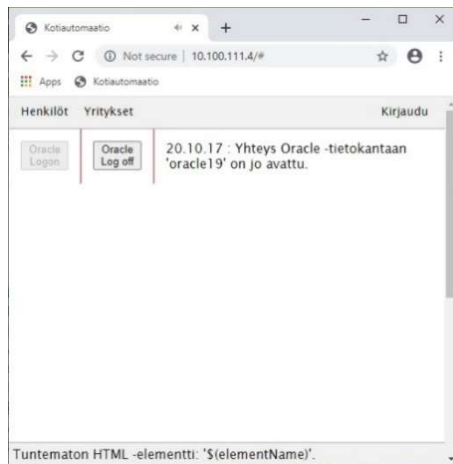


Kuvio 14: Login



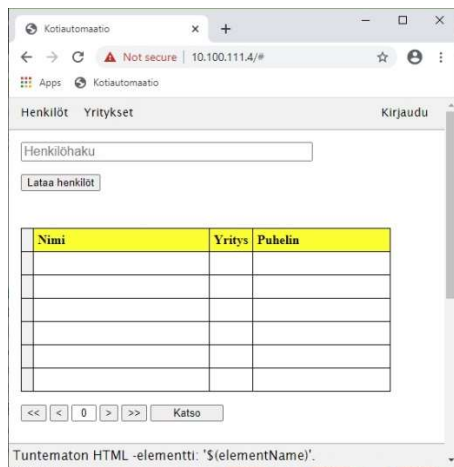
Kuvio 15: Log Out

Jos yhteys on jo aiemmin avattu, sivusto ilmoittaa siitä käyttäjälle (Kuvio 18).



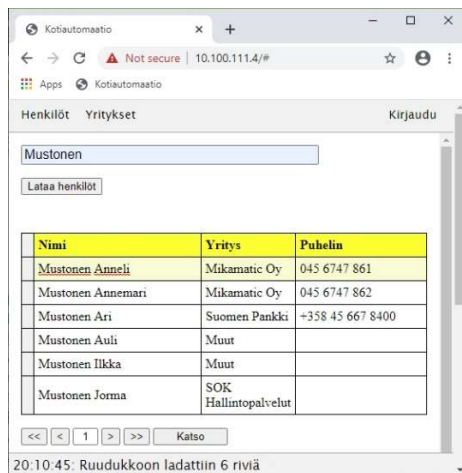
Kuvio 16: Kirjaudu - sivu, yhteys on jo avattu

Henkilöt - sivulle mentäessä avautuu ensin tyhjä taulukko (Kuvio 19).



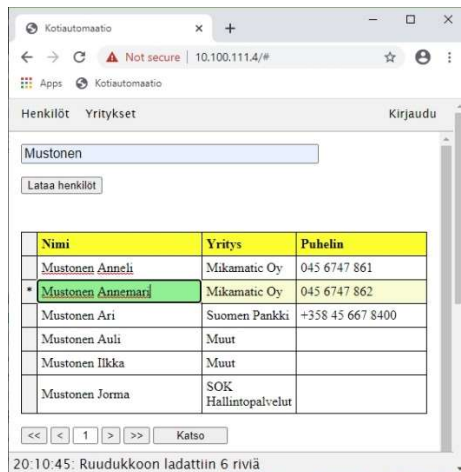
Kuvio 17: Tyhjä henkilöt-sivu

Kun käyttäjä kirjoittaa jonkin hakuehdon hakukenttään ja painaa Lataa henkilöt - painiketta, listaan tulee näkyviin hakuehtoja vastaavia henkilöitä. Jos henkilöitä on enemmän kuin kuusi, listaa voi kelata taulukon alareunan nuolinäppäimillä (Kuvio 20).



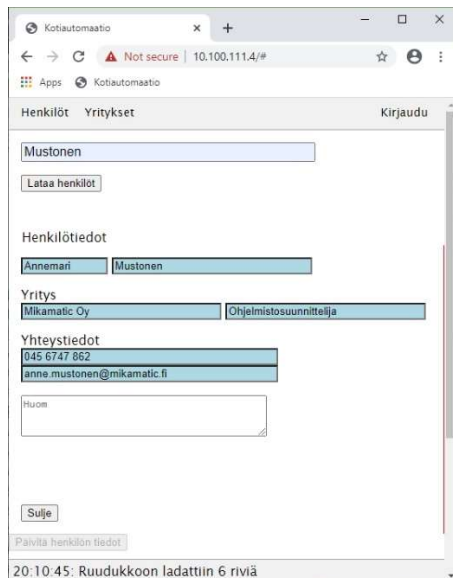
Kuvio 18: Täysi Henkilöt - lista

Kun käyttäjä klikkaa yksittäistä taulukon riviä, hänellä on mahdollisuus muuttaa ja tallentaa henkilön tietoja tietokantaan jo tässä vaiheessa (Kuvio 21).



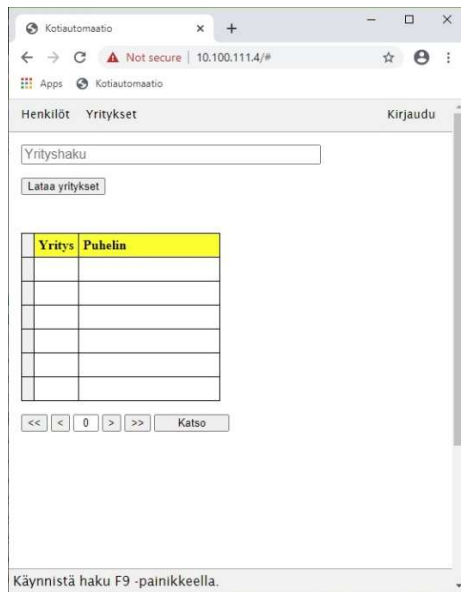
Kuvio 19: Henkilöt - lista edit - tilassa

Kun käyttäjällä on yksi taulukon rivi valittuna, hän voi klikata Katso - nappia, joka avaa yksityiskohtaiset tiedot kyseisestä henkilöstä. Kaikki kentät ovat muokattavissa ja niiden tiedot päivittyvät tietokantaan. Sulje - nappi palauttaa käyttäjän takaisin henkilöt - listaan (Kuvio 22).

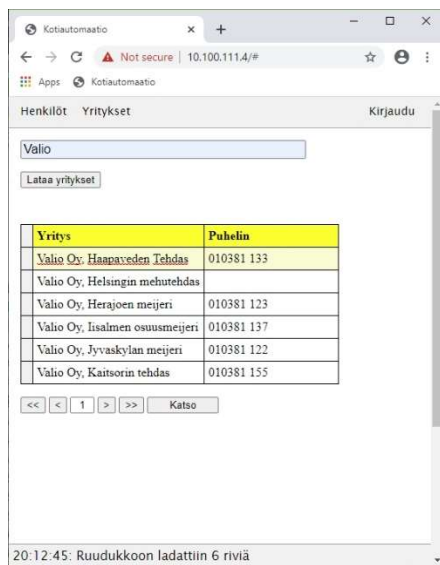


Kuvio 20: Henkilön täydet tiedot -sivu

Yritykset - sivu avaa aluksi samantapaisen tyhjän listan kuin henkilöt -sivukin ja siihen pätevät samat toiminnallisuudet (Kuviot 23 ja 24).



Kuvio 21: Yritykset - sivu



Kuvio 22: Täysi yritykset - sivu

Yksittäisen yrityksen sivu. Kun käyttäjä editoi jotain kentistä, alun perin peitettynä ollut Päivitä yrityksen tiedot - nappi aktivoituu ja sen avulla tiedot talletetaan tietokantaan. Tämä toimii myös henkilötiedot - sivulla (Kuvio 25).

Kotiautomaatio

Not secure | 10.100.111.4/#

Apps Kotiautomaatio

Henkilöt Yritykset Kirjaudu

Valio

Lataa yritykset

Yritystiedot

Valio Oy

Yhteystiedot

010381 123

testi@testiinen.fi

Huom

Sulje

Päivitä yrityksen tiedot

20:13:7: Ruudukkoon ladattiin 6 riviä

Kuvio 23: Yritystiedot päivittymässä

4.2 Jatkokehitys

Sivuston jatkokehitys on välttämättöntä, jos idea on tarkoitus jossain vaiheessa esitellä myös asiakkaillemme. Heidän tarpeisiinsa sivustoa täytyisi muokata huomattavasti tiedonhaun osalta, mutta perusteet pysyisivät samana. Myös omaan käyttöön sivustoa on kehitettävä edelleen, sillä puhelun soittaminen ja sähköpostin lähetys sen kautta olisi erittäin hyödyllistä. Tällä hetkellä sivustolla ei ole erillistä sisäänkirjautumista, mikä hankaloittaa sen käyttöä esimerkiksi ulkoisista verkoista. Sivuston ulkoasu kaipaava myös pientä viimeistelyä, mutta se on kuitenkin tarpeeksi hyvä ja selkeä omaan käyttöömme. Eräs mahdollinen kehityssuunta on myös toiminnallisuuden laajentaminen esimerkiksi tuntikirjausten tekoon, joka tapahtuu tällä hetkellä Yritysrekisterin kautta.

4.3 Arviointi

Opinnäytetyön suunniteltu aikataulu ei pitänyt ollenkaan. Suurin syy tähän oli koronan takia äkillisesti lisääntynyt työmäärä, sillä tein samaan aikaan myös tavallisia töitä yrityksessämme. Meillä on kiinteä laskutus, joten asiakkaamme pyrkivät kiristyvässä taloustilanteessa keskittämään töitä meille.

Sivuston kehitys oli minulle mielenkiintoinen ja hyödyllinen projekti, jossa pääsin verran käyttämään vasta opittuja mobiiliohjelmointitaitojani. Paljon täytyi kuitenkin opiskella lisää opinnäytetyön edetessä, ja sen kehitykseen liittyi paljon yrityksiä ja erehdyksiä. Siitä tuli omasta mielestäni mukavan näppärä pieni sivusto, jolla on käyttöä nyt ja tulevaisuudessa sekä omassa yrityksessämme että mallina mahdollisille tuleville asiakkaille.

Lähteet

Sähköiset

A brief overview of Bitbucket 2020, Atlassian BitBucket, Viitattu 16.12.2020,
<https://bitbucket.org/product/guides/getting-started/overview#a-brief-overview-of-bitbucket>

Borho 2020, About TortoiseHG, TortoiseHG, Viitattu 16.12.2020,
<https://tortoisehg.bitbucket.io/about.html>

Client-Server Overview 2020, MDN Web Docs, Viitattu 15.12.2020,
https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview

CSS Tutorial 2020, Tutorialspoint, Viitattu 14.12.2020,
<https://www.tutorialspoint.com/css/index.htm>

Embarcadero 2016, Language Overview, Viitattu 03.08.2020,
http://docwiki.embarcadero.com/RADStudio/Rio/en/Language_Overview

HTML: HyperText Markup Language 2019, MDN Web Docs, Viitattu 24.04.2020,
<https://developer.mozilla.org/fi/docs/Web/HTML>

Ionic Framework 2020, Ionic Framework, Viitattu 16.12.2020,
<https://ionicframework.com/docs>

Marvel 2020, Everything you need to bring ideas to life, Viitattu 20.03.2020,
<https://marvelapp.com/why-marvel>

Microsoft 2020, Learn to code with Visual Studio Code, Viitattu 15.12.2020
<https://code.visualstudio.com/learn>

Oracle 2020, Introduction to Oracle Database, Viitattu 16.09.2020,
<https://docs.oracle.com/en/database/oracle/oracle-database/21/cncpt/introduction-to-oracle-database.html#GUID-A42A6EF0-20F8-4F4B-AFF7-09C100AE581E>

PluralSight 2020, Strings, Viitattu 09.05.2020, <https://www.javascript.com/learn/strings>

PL/SQL - Overview 2020, Tutorialspoint, Viitattu 14.12.2020,
https://www.tutorialspoint.com/plsql/plsql_overview.htm

Sarja 2006, Relaatietietokanta, Verkkopedagogi,
<https://verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku0375c6.html?C>

SPA (Single-page application) 2020, MDN Web Docs, Viitattu 15.12.2020,
<https://developer.mozilla.org/en-US/docs/Glossary/SPA>

SQL - Overview 2020, Tutorialspoint, Viitattu 14.12.2020,
<https://www.tutorialspoint.com/sql/sql-overview.htm>

Typescript 2020, What is TypeScript?, Viitattu 16.12.2020, <https://www.typescriptlang.org/>

What is a web server? 2020, MDN Web Docs, Viitattu 15.12.2020,
https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

Work easier work faster 2020, Mercurial, Viitattu 16.12.2020, <https://www.mercurial-scm.org/>

Kuviot

Kuvio 1: TortoiseHg-käyttöliittymä	12
Kuvio 2: Kommentoitua koodia	13
Kuvio 3: Kovakoodattu hakuehto	16
Kuvio 4: Esimerkki PopUp - ikkunasta	17
Kuvio 5: Toinen esimerkki PopUp - ikkunasta	17
Kuvio 8: Case - lause esimerkki.....	18
Kuvio 9: Sivun koodivirheen jälkeen.....	19
Kuvio 10: Kohta, jossa virhe oli.....	19
Kuvio 11: Virhe korjattu	19
Kuvio 12: Valkoinen alue	20
Kuvio 13: Korostetun rivin lisäys poisti ongelman.....	20
Kuvio 14: HTMLInput - virhe.....	21
Kuvio 15: Kirjautu-sivu	22
Kuvio 16: Login.....	22
Kuvio 17: Log Out.....	23
Kuvio 18: Kirjautu - sivu, yhteys on jo avattu.....	23
Kuvio 19: Tyhjä henkilöt-sivu	24
Kuvio 20: Täysi Henkilöt - lista	24
Kuvio 21: Henkilöt - lista edit - tilassa	25
Kuvio 22: Henkilön täydetyt tiedot -sivu	25
Kuvio 23: Yritykset - sivu	26
Kuvio 24: Täysi yritykset - sivu	26
Kuvio 25: Yritystiedot päivittymässä.....	27