

Miika Forma

SOVELLUS TUNTIKIRJAUKSEEN

Tietojenkäsittelyn koulutusohjelma
2019

SOVELLUS TUNTIKIRJAUKSEEN

Forma, Miika
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Lokakuu 2019
Sivumäärä: 39

Asiasanat: työajanseuranta, laskutus, sovellus, ohjelmointi

Tämän opinnäytetyön tarkoituksena oli tuottaa vuonna 2014 perustetulle IT-alan palveluita tarjoavalle Codecontrol Oy:lle laskutusta varten tuntien ja matkalaskujen kirjaamisen mahdollistava sovellus. Näin yrityksen on helpompi laskuttaa asiakkaita.

Digitaalisessa työajanseurannassa on helpompi kirjata ja seurata työaikoja. Digitaalisessa muodossa olevat kirjaukset säilyvät paremmin ja ovat helpommin saatavilla myös pitkänkin ajan jälkeen.

Sovelluksen tekemisessä käytettiin mm. seuraavia tekniikoita ja ohjelmointikieliä: ASP.NET MVC, C#, Entity Framework HTML, CSS sekä JavaScript. Sovelluksen testaus suoritettiin yksikkötesteillä sekä hyväksyntätestauksilla.

SOFTWARE FOR TIME TRACKING

Forma, Miika

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Business Information Systems

October 2019

Number of pages: 39

Keywords: time tracking, billing, software, programming

The purpose of this thesis was to produce work hour and travel expense tracking software for IT consultant company Codecontrol Oy which was founded in 2014. This makes billing clients easier for the company.

Digital work hour tracking makes it easier to log and follow working hours. Time tracking entries last better when they are stored digitally, and they will be easily accessible even after long period of time.

Some of the techniques and languages used in the production of this software were ASP.NET MVC, C#, Entity Framework, HTML, CSS and JavaScript. Testing of the software was done with unit and acceptance tests.

SISÄLLYS

1	JOHDANTO	6
2	MÄÄRITTELY JA SUUNNITTELU	7
3	KÄYTETYT TEKNIIKAT	8
3.1	Backend-tekniikat	8
3.1.1	ASP.NET MVC	8
3.1.2	C#	9
3.1.3	ASP.NET Identity	9
3.1.4	Entity Framework	10
3.1.5	Elmah.....	10
3.2	Frontend-tekniikat	11
3.2.1	HTML.....	11
3.2.2	CSS.....	11
3.2.3	JavaScript.....	12
3.2.4	jQuery	13
3.2.5	Bootstrap.....	13
3.3	Muut tekniikat	14
3.3.1	MySQL	14
3.3.2	RabbitMQ	15
3.3.3	Robot Framework	15
4	TOTEUTUS.....	16
4.1	Käyttöliittymä	16
4.2	Työntekijät	18
4.3	Asiakkaat	19
4.4	Projektit	20
4.5	Työtunnit	22
4.5.1	Tuntikirjaukset	22
4.5.2	Työaikaseuranta	26
4.6	Matkalaskut.....	27
4.7	Raportit	29
4.8	Käyttäjät.....	31
4.9	Asetukset	32
5	LOGIEN TEKEMINEN	34
5.1	Loki käytön seurantaan.....	34
5.2	Virheloki.....	35
6	TESTAUS.....	35
7	POHDINTA.....	37

1 JOHDANTO

Tuntien kirjaaminen on aina ollut erittäin tärkeää, jotta on pystytty seuraamaan työaikoja ja niihin on pystytty palaamaan tarvittaessa. Työaikakirjanpitovelvoite määritellään työaikalaisissa (Tyosuojelu.fi www-sivut 2019). Seurannan avulla esimerkiksi viranomaisten on helpompi tarkistaa, että lomat ja ylityöt ovat hoidettu lainmukaisesti. Tehdyt työtunnit ovat myös laskutuksen perusteena. Tuntikirjaus mahdollistaa totuudenmukaisen laskutuksen tehdystä työstä, koska muutoin laskutus perustuisi arvailuun (Harvest www-sivut 2019).

Kirjaamisen avulla voidaan seurata, tekeekö projekti rahaa, vai kuluttaako se sitä. Sen avulla on mahdollista pysyä ajan tasalla projektin aikataulun suhteen sekä huolehtia työntekijöiden jaksamisesta. (Harvest www-sivut 2019.)

Nykyaikana suurin osa tuntien kirjaamisista on digitaalista ja monet järjestelmät ovat myös mobiililaitteilla toimivia. Tämä helpottaa tuntien kirjaamista sekä palkanlaskentaa. Kaikki kirjaukset ovat keskitetty yhteen paikkaan ja ovat helposti haettavissa pitkänkin ajan jälkeen.

Tässä työssä käydään läpi Codecontrol Oy:lle tekemääni tuntikirjaus sovellusta, jota yritys tarvitsee laskutusta varten. Lisäksi sovelluksella on mahdollista kirjata matkalaskuja ja muita matkakuluja. Sovellus on toteutettu yrityksen tarpeiden mukaan.

2 MÄÄRITTELY JA SUUNNITTELU

Aikaisemmin käytössä on ollut vanhalle ASP.NET MVC 4 -pohjalle tehty sovellus. Uudistuksen myötä tarkoituksena on modernisoida sovellusta ja korjata vanhassa olleita ongelmia. Vanhassa sovelluksessa käytössä on ollut MySQL-tietokanta ja tunnistautumiseen on käytetty ASP.NET Membership -tekniikkaa.

MySQL-tietokantaa tullaan käyttämään myös uudessa sovelluksessa, koska se sopii käyttötarkoituksiin ja on ilmainen. Suoraan vanhaa tietokantaa ei kuitenkaan voi käyttää, koska siihen tulee väkisinkin muutoksia. Tunnistautumisessa siirrytään käyttämään uudempaa ASP.NET Identity -tekniikkaa, joka muokkaa kannan rakennetta. Myös uudet ominaisuudet tulevat lisäämään ja poistamaan kenttiä sekä mahdollisesti kokonaisia tauluja. Näin ollen on kannattavaa tehdä uusi kanta, johon vanha data lisätään migraatioiden avulla.

Uusi tekniikka salaa käyttäjien salasanat eri tavalla, joten sovellukseen täytyy ohjelmoida tuki myös vanhalle salaukselle. Salasana vahvistetaan vanhalla salauksella ja tämän jälkeen se salataan uudella tekniikalla, jota tullaan käyttämään kaikissa sen jälkeisissä kirjautumisissa.

Käyttöliittymän toteuttamiseen valikoitui jQuery ja Bootstrap, joiden avulla sovellukseen saadaan tuki myös mobiilille, ilman omia suuria CSS-muutoksia. jQuery-kirjaston avulla käyttöliittymää saadaan muokattua monipuolisesti käyttäjän selaimessa. Näin päästään eroon mm. koko sivun uudelleen latauksista, kun halutaan päivittää jonkin sivulla olevan elementin sisältö. Bootstrapin avulla saadaan helposti mm. piilotettua sisältöä mobiililaitteilta, vaikka kyseinen sisältö näytettäisiinkin isommilla näytöillä.

3 KÄYTETYT TEKNIIKAT

3.1 Backend-tekniikat

Backend-tekniikat ovat palvelimen päähän tulevan koodin tekemiseen käytettyjä tekniikoita. Se on osa ohjelmaa, jota käyttäjät eivät näe, mutta on silti tärkeä osa ohjelman toimintaa.

3.1.1 ASP.NET MVC

Model View Controller (MVC) on suunnittelumalli, jonka avulla erotetaan käyttöliittymä, datamalli ja ohjelman logiikka toisistaan. (Microsoft [www-sivut](#), 2019a.)

Malli (Model) -luokkia käytetään ohjelman tietojen ylläpitämiseen. Niihin haetaan ja niistä tallennetaan tietoja pysyviin tietovarastoihin, kuten tietokantoihin. Ohjain (Controller) -luokat vastaavat siitä, mitä tietoja malli luokat sisältävät. Näkymät (Views) sen sijaan vastaavat siitä, minkälaisena mallien sisältämät tiedot näytetään käyttäjille. (Tutorialspoint [www-sivut](#), 2019a.)

Ohjaimet ovat yksi tärkeimmistä ASP.NET MVC -ohjelman osista. Ne vastaanottavat HTTP-pyyntö ja lähettää oikean mallin näkymälle prosessoitavaksi. HTTP-pyyntö on selaimen lähettämä viesti, jonka rakenne noudattaa HTTP-protokollaa. Tämän jälkeen ohjain palauttaa prosessoidun näkymän takaisin HTTP-pyyntö lähettäjälle. Ohjaimet voivat myös näkymien sijasta tehdä jonkin muun toiminnon, kuten esimerkiksi ohjata toiseen ohjaimen toimintoon. (Tutorialspoint [www-sivut](#), 2019b; Tutorialspoint [www-sivut](#), 2019c.)

Näkymissä muokataan mallien sisältämä tieto siihen ulkoasuun, jolta se näyttää käyttäjien selaimissa. (Tutorialspoint [www-sivut](#), 2019c.)

3.1.2 C#

C# on tyyppiturvallinen ohjelmointikieli, jonka avulla on helppo ohjelmoida turvallisia ja vakaita sovelluksia. Kyseistä ohjelmointikieltä on käytetty myös tämän sovelluksen palvelinpään koodin kirjoittamiseen. (Microsoft www-sivut, 2019b.)

C#:n kieliasu on selkeää englannin kieltä, joten sitä on huomattavasti helpompi ymmärtää, kuin esimerkiksi C++-kieltä. (Microsoft www-sivut, 2019c.)

```
C#  
  
// A Hello World! program in C#.  
using System;  
namespace HelloWorld  
{  
    class Hello  
    {  
        static void Main()  
        {  
            Console.WriteLine("Hello World!");  
  
            // Keep the console window open in debug mode.  
            Console.WriteLine("Press any key to exit.");  
            Console.ReadKey();  
        }  
    }  
}
```

Kuva 1. C# Hello World esimerkki (Microsoft www-sivut, 2019c.)

3.1.3 ASP.NET Identity

Aikoinaan palveluihin kirjaututtiin sisään antamalla käyttäjänimi sekä salasana, joita oli käytetty palveluun rekisteröityessä. Nykyään tämä ei enää pidä paikkaansa, vaan on myös paljon muita tapoja, joilla voidaan kirjautua sisään. Ohjelmia voidaan kehittää niin, että palveluun voidaan kirjautua hyödyntäen toiseen palveluun rekisteröitymistä. Tällaisia palveluja ovat esimerkiksi Facebook sekä Twitter. (Microsoft www-sivut, 2019d.)

ASP.NET Identity kehitettiin pitäen seuraavia tavoitteita mielessä:

- Olisi vain yksi hallintajärjestelmä, jota voitaisiin käyttää kaikissa ASP.NET -ohjelmistokehyksissä.

- Tietokannan rakennetta on helppo muokata niin, että siihen voidaan esimerkiksi lisätä käyttäjän syntymäpäivä.
- Eri tallennusmekanismien käyttöönottoaminen on helppoa.
- Helpottaa yksikkötestien tekemistä niihin osiin koodia, jotka käyttävät ASP.NET Identity -palvelua.
- Roolijärjestelmä, jonka avulla käyttäjien pääsyä eri alueille voidaan rajoittaa.
- Tuki sosiaalisten palveluiden kirjautumisjärjestelmille kuten esimerkiksi Microsoft-, Facebook-, Twitter- sekä Google-tileille.
- Mahdollisuus OWIN-integraatioon. OWIN on standardi palvelimen ja web-sovellusten väliseen kommunikointiin. Tämä mahdollistaa sen, että FormsAuthentication-autentikoinnin sijasta voidaan käyttää OWIN CookieAuthentication -autentikointia.
- Palvelu toimitetaan NuGet-pakettina, joka mahdollistaa uusien ominaisuuksien sekä bugi korjauksien toimittamisen kehittäjille ketterästi.

(Microsoft www-sivut, 2019d.)

3.1.4 Entity Framework

Entity Framework on ohjelmointikehys ORM:lle. Sen tehtävänä on yhdistää relaatio-tietokanta luokkiin sekä niiden ominaisuuksiin. Sen avulla saadaan myös eroteltua tietokantayhteyden käytettävä logiikka ohjelman logiikasta. (Musso 2016.)

3.1.5 Elmah

ELMAH on virheiden kirjaamiseen sekä käsittelyyn käytettävä ilmainen vapaaseen lähdekoodiin perustuva kirjasto. Siihen on sisäänrakennettu toiminto, jonka avulla kirjattuja virheitä voidaan katsoa verkkosivun kautta. Siinä on myös tuki RSS-syötteille sekä virheet voidaan ladata CSV-tiedostona. (Mitchell 2009.)

3.2 Frontend-tekniikat

Frontend-tekniikat ovat tekniikkoja, joita käytetään käyttäjälle näkyvän käyttöliittymän tekemiseen. Niiden avulla tehdään käyttäjän selaimessa näkyvä käyttöliittymä.

3.2.1 HTML

HTML eli HyperText Markup Language on yksi nettisivujen tärkeimmistä rakennuspalikoista. Sen keskittyy nettisivun rakenteen kuvaamiseen, kun taas muita nettisivujen kehitykseen käytettyjä tekniikoita käytetään pääasiallisesti ulkoasuun ja toiminnallisuuteen. (wbamberg 2019.)

HyperText viittaa linkkeihin, joiden kautta selataan sivulta toiselle. Näiden linkkien avulla voidaan siirtyä sivulta toiselle yhden nettisivuston sisällä tai niiden avulla voidaan myös siirtyä nettisivustolta toiselle. (wbamberg 2019.)

Itse rakenne tehdään elementeillä, joita ovat esimerkiksi ”<head>”, ”<title>”, ”<body>”, ”<footer>” sekä monet muut. Elementti tehdään niin, että laitetaan jokin teksti ”<” ja ”>” -merkkien väliin. Kyseinen tekstin kirjainkoolla ei ole väliä, eli se saa sisältää pieniä sekä isoja kirjaimia. (wbamberg 2019.)

3.2.2 CSS

Cascading Style Sheets eli CSS on kieli, jonka avulla pystytään muokkaamaan HTML-koodilla tehtyjen nettisivujen ulkoasua. Sen avulla kerrotaan, miten HTML-elementit näytetään näyttöllä. Eli sen avulla siis määritetään esimerkiksi mitä värejä ja fontteja sivulla käytetään sekä miten kaikki on sijoiteltu. (Morris 2018; SphinxKnight 2019.)

CSS tyylejä voidaan määritellä suoraan HTML-dokumentin sisään (Kuva 2), tai niitä voidaan kirjoittaa erilliseen tiedostoon ja kutsua HTML-dokumentissa (Kuva 3). CSS-tyylimäärittelyjä on myös mahdollista kirjoittaa suoraan HTML-elementteihin.

```
<head>
<style>
body {
  background-color: green;
}
</style>
</head>
```

Kuva 2. CSS-tyylimäärittelyjä HTML-dokumentin sisällä.

```
<head>
<link rel="stylesheet" type="text/css href="samk-opparistyle.css">
</head>
```

Kuva 3. CSS-tyylimäärittelyjen sisällytys ulkoisesta tiedostosta.

```
<body style="background-color: black;">
<h1 style="color: white;">SAMK Opinnäytetyö</h1>
</body>
```

Kuva 4. CSS-tyylimäärittelyjä HTML-elementeissä.

3.2.3 JavaScript

JavaScript on dynaaminen kieli, jonka avulla voidaan lisätä nettisivuille dynaamista toimintaa. Vaikka se tunnetaan parhaiten nettisivujen skriptauskielenä, käytetään sitä myös monessa muussa paikassa kuin selainympäristöissä (esim. NodeJS). (Mills 2019.)

```

1  getEmployees = () => {
2      var request = new XMLHttpRequest();
3      request.open('GET',
4          'http://dummy.restapiexample.com/api/v1/employees',
5          true);
6
7      request.onload = function() {
8          if (this.status >= 200 && this.status < 400) {
9              // Onnistui, työntekijät haettu
10             var data = JSON.parse(this.response);
11         } else {
12             // Yhteys onnistui, mutta tapahtui jokin virhe
13         }
14     };
15
16     request.send();
17 };

```

Kuva 5. JavaScript-esimerkki, jossa haetaan kaikki työntekijät REST-rajapinnasta.

3.2.4 jQuery

jQuery on JavaScript-kirjasto, joka sisältää suuren määrän ominaisuuksia. Sen avulla pystyy helposti muokkaamaan ja selaamaan HTML-dokumentteja. Myös muun muassa Ajax-kutsujen tekeminen on huomattavasti helpompaa jQuery-kirjaston avulla kuin natiivilla JavaScriptillä kirjoitettuna (vertaa Kuva 5 ja Kuva 6). (jQuery www-sivut, 2019.)

```

1  getEmployees = () => {
2      var jqxhr = $.get("http://dummy.restapiexample.com/api/v1/employees", function() {
3          // Onnistui, työntekijät haettu
4          var data = JSON.parse(this.response);
5      })
6      .fail(function() {
7          // Tapahtui jokin virhe
8      });
9  };

```

Kuva 6. jQuery esimerkki, jossa haetaan kaikki työntekijä REST-rajapinnasta.

3.2.5 Bootstrap

Bootstrap on avoimeen lähdekoodiin perustuva ohjelmistokehys, jonka avulla saadaan tehtyä vuorovaikutteisia sivuja, jotka toimivat useilla eri selaimilla. Bootstrapin

kehityksen aloitti muutama Twitterin työntekijä ja sen tavoitteena oli yhdenmukaisuus. (Caliman 2015.)

Kun käyttää Bootstrap-kirjaston sisältämiä pohjia ja luokkia, säästää huomattavasti aikaa, kun ei tarvitse itse tehdä monimutkaisia CSS-sääntöjä. HTML-elementeille annetaan luokkia ja Bootstrap-kirjasto hoitaa taustalla elementtien oikein näyttämisen eri selaimilla. (Caliman 2015.)

3.3 Muut tekniikat

3.3.1 MySQL

MySQL on tietokannan hallintajärjestelmä, jonka avulla voidaan hallinnoida relaatio-tietokantoja. Se on Oraclen tukema julkiseen lähdekoodiin perustuva ohjelmisto, joten sitä voidaan käyttää ilmaiseksi. (MySQLTTUTORIAL [www-sivut](#), 2019)

Ensimmäinen versio julkaistiin vuonna 1995, ja se tehtiin alun perin henkilökohtaiseen käyttöön. Muutaman vuoden aikana siitä kuitenkin kehittyi maailman käytetyin julkisen lähdekoodin relaatiotietokantaohjelmisto. Tammikuussa 2018 Sun Microsystems osti MySQL:n ja pian sen jälkeen Oracle osti Sun Microsystemsin. Osa alkuperäisistä kehittäjistä ei pitänyt MySQL omistajuuden siirtymisestä Oraclelle. Tämän seurauksena he jatkoivat MySQL:n kehittämistä ja julkaisivat sen uudella nimellä MariaDB vuonna 2009. Molemmat ohjelmistot ovat julkiseen lähdekoodiin perustuvia, mutta MySQL:n uudet lisäykset toimitetaan suurina palasina pitkällä viiveellä. Sen sijaan MariaDB kehitys on täysin avointa ja kaikkia kehityspäätöksiä voidaan seurata sekä kommentoida kehityksen yhteydessä. (MySQLTTUTORIAL [www-sivut](#), 2019; Seravo [www-sivut](#), 2015; Goel 2019.)

MariaDB ja MySQL eroavat toisistaan myös lisensoinnin puolesta. MariaDB on lisensoitu GPLv2-lisenssillä. MySQL sen sijaan omaa kaksi eri lisenssivaihtoehtoa, joista toinen (GPLv2) on tarkoitettu yhteisölle ja toinen yrityksille. MariaDB tarjoaa kaikki ominaisuudet, kun taas MySQL:n yhteisöversiosta puuttuu hyödyllisiä ominaisuuksia, jotka parantavat suorituskykyä. (Shay 2018.)

3.3.2 RabbitMQ

RabbitMQ on yksi suosituimmista avoimen lähdekoodin viestinvälittäjäsovelluksista. Sen avulla voidaan muun muassa vähentää paljon aikaa vievien tehtävien kuormitusta ulkoistamalla kyseiset tehtävät palvelimille, joilla ei ole muuta työtä. (RabbitMQ [www-sivut](#), 2019; Johansson 2019.)

Viestinvälittäjää on myös hyvä käyttää silloin, kuin lähetetään esimerkiksi sähköposteja asiakkaille. Ensiksi ohjelma muodostaa lähetettävän viestin ja lähettää sen RabbitMQ:n viestijonoon. Tämän jälkeen alkuperäisen ohjelman ei tarvitse enää välittää kyseisestä viestistä, vaan RabbitMQ pitää huolen sen perille toimittamisesta. Seuraavaksi viesti välitetään oikealle viestin käsittelijälle. Viestin käsittelijä käsittelee viestin ja se poistetaan viestijonosta. RabbitMQ voidaan kuitenkin esimerkiksi sähköpostien tapauksessa konfiguroida niin, että viestin toimitusta yritetään uudestaan, jos käsittelijä epäonnistuu sähköpostin toimituksessa. (Johansson 2019.)

3.3.3 Robot Framework

Robot Framework on Nokia Networksin kehittämä ohjelmistokehys, jonka avulla voidaan tehdä muun muassa hyväksyntätestauksia. Sen avulla voidaan testata ohjelman toimintaa päästä päähän eli selaimelta palvelimelle ja takaisin. Näin voidaan testata mm. käyttäjän näkymien toimivuus. (RobotFramework [www-sivut](#), 2019.)

Yksikkötestit ja integraatiotestit testaavat ohjelman sisäistä toimintaa. Näitä testejä voi olla hankala ymmärtää ilman ohjelmointitaustaa. Robot Frameworkin hyväksyntätesteissä käytetään selkeää englannin kieltä, joten niitä on asiakkaankin helppo ymmärtää, vaikka hänellä ei olisikaan ohjelmointitaustaa (Kuva 7). (Saldanha 2014.)

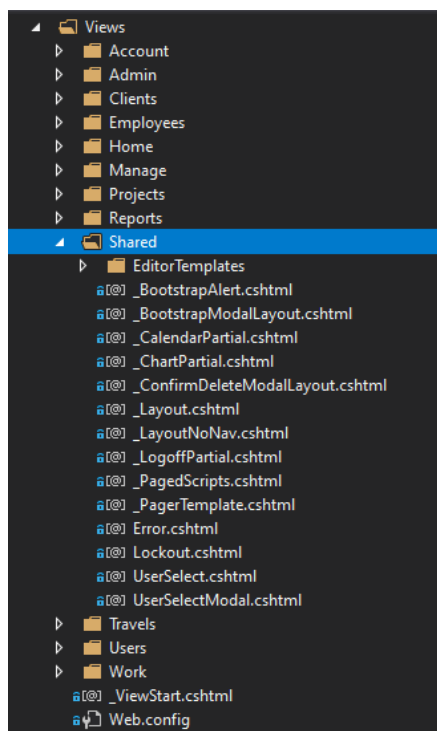
```
CcTracker-set-admin.robot x
1
2  *** Settings ***
3  Resource      CcTracker_common_keywords.robot
4  Documentation  A test suite for CcTracker log in and log out with valid and invalid user name and password
5  Suite Setup    Go to loginpage
6  Suite Teardown Close All Browsers
7
8  *** Variables ***
9  ${VALID_USERNAME}    testi
10 ${VALID_PASSWORD}    testipassu
11 ${INVALID_USERNAME}  abc
12 ${INVALID_PASSWORD}  def
13
14 *** Test Cases ***
15 Valid user logins
16     When User logins "${VALID_USERNAME}" "${VALID_PASSWORD}"
17     Then Work page user selection is shown
18
19 User logs out
20     When User logout "${VALID_USERNAME}"
21
22 Invalid user logins
23     When User logins "${INVALID_USERNAME}" "${INVALID_PASSWORD}"
24     Then Invalid login is shown
25
26 Empty user logins
27     When User logins "" ""
28     Then Empty error is shown
```

Kuva 7. Esimerkki hyväksyntätestien testitapauksista CcTracker-kirjautumisesta.

4 TOTEUTUS

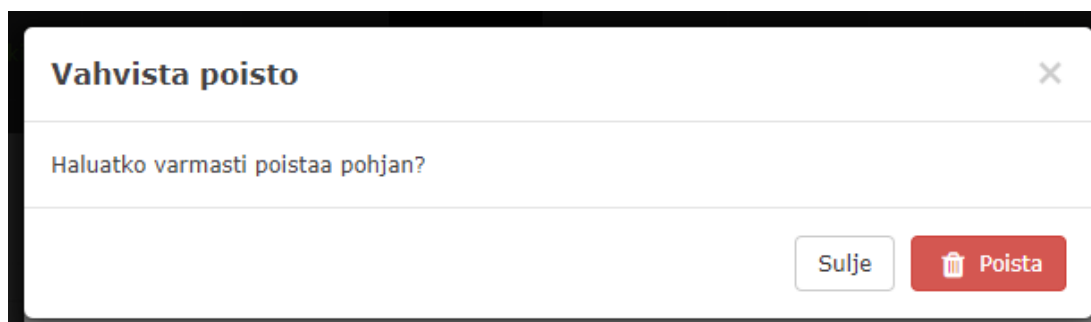
4.1 Käyttöliittymä

Käyttöliittymän tekemisessä on käytetty ASP.NET Razor -syntaksia. Ulkoasussa on hyödynnetty Bootstrap-komponenttikirjastoa, jonka avulla on nopeaa tehdä vuorovai-
kutteisia sivuja. Rakenteen muokkauksiin on pääsääntöisesti käytetty jQuery-kirjastoa, mutta useilla sivuilla on myös käytetty KnockoutJS-kirjastoa.



Kuva 8. Razor näkymiä, joita CcTrackerissä on käytetty.

Suurimmassa osassa ohjelmaa käytetään Modal-ikkunoita, jotka ovat sisällytetty Bootstrap komponenttikirjastoon JavaScript lisäosana. Nämä Modal-ikkunat ovat dialogeja, jotka aukeavat sivun päälle (Kuva 9). Kyseisissä ikkunoissa voidaan sekä näyttää, että täyttää tietoja. Niitä käytetäänkin tietojen lisäämiseen, kuten esimerkiksi uuden käyttäjän tietojen täyttämiseen, mutta myös poistamisen vahvistamiseen (Kuva 9).



Kuva 9. Poiston vahvistaminen.

Ohjelmassa on tuki sekä suomen että englannin kielelle. Oletuskielenä on suomi, mutta kielen voi vaihtaa jokaisella sivulla oikeassa alareunassa näkyvästä pudotusvalikosta (Kuva 10).

The screenshot shows the CodeControl web application interface. The top navigation bar includes the CodeControl logo and several menu items: Employees, Clients, Projects, Work hours, Travel expenses, Reports, Users, and Settings. A 'New employee' button is visible. Below the navigation bar is a search input field labeled 'Search Q'. The main content area displays a table of employees with columns for Lastname, Firstname, Email, Phone number, and Actions. The table contains two rows: one for 'Forma' (Miika) and one for 'Inaktiivinen' (Työntekijä). The 'Actions' column for each row contains three icons: a grey eye, a green gear, and a red trash can. At the bottom of the page, there is a copyright notice '© 2019 - Codecontrol' and a language dropdown menu set to 'English'.

Kuva 10. Englannin kieli valittuna.

4.2 Työntekijät

The screenshot shows the CodeControl web application interface with the language set to Finnish. The top navigation bar includes the CodeControl logo and menu items: Työntekijät, Asiakkaat, Projektit, Työtunnit, Matkalaskut, Raportit, Käyttäjät, and Asetukset. A 'Uusi työntekijä' button is visible. Below the navigation bar is a search input field labeled 'Haku Q'. The main content area displays a table of employees with columns for Sukunimi, Etunimi, Sähköposti, Puhelinnumero, and Toiminnot. The table contains two rows: one for 'Forma' (Miika) and one for 'Inaktiivinen' (Työntekijä). The 'Toiminnot' column for each row contains three icons: a grey eye, a green gear, and a red trash can. At the bottom of the page, there is a copyright notice '© 2019 - Codecontrol' and a language dropdown menu set to 'Suomi'.

Kuva 11. Työntekijät-näkymä.

”Työntekijät”-sivulla nimensä mukaisesti näytetään lista työntekijöistä. Sieltä päästään lisäämään uusia työntekijöitä (Kuva 12), muokkaamaan työntekijöiden tietoja, poistamaan työntekijöitä sekä halutessaan työntekijän voi laittaa inaktiiviseen tilaan. Inaktiivisessa tilassa olevaa työntekijää ei näytetä missään muualla kuin työntekijät listassa, eli se toimii lähes samalla tavalla kuin poistaminen. Etuna on kuitenkin se, että kyseinen työntekijä voidaan uudelleen aktivoida myöhemmin vain yhdellä klikkauksella ilman, että kaikkia tietoja tarvitsee lisätä uudestaan tai käyttäjän historia olisi kadonnut.

Uusi työntekijä

Etunimi

Sukunimi

Sähköposti

Puhelinnumero

Katuosoite

Kaupunki

Postinumero

Maa

Sulje Tallenna

Kuva 12. Työntekijän lisäys.

4.3 Asiakkaat

CodeControl Työntekijät **Asiakkaat** Projektit Työtunnit Matkalaskut Raportit Käyttäjät Asetukset Admin

Uusi asiakas Uusi yhteyshenkilö

Haku Q

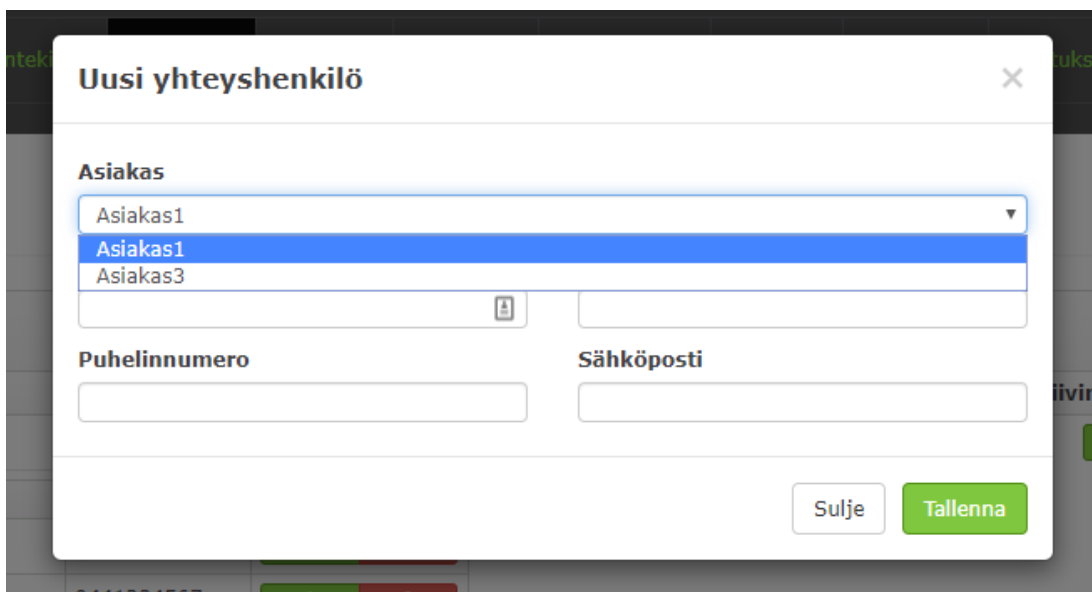
Nimi	Katuosoite	Kaupunki	Postinumero	Aktiivinen									
Asiakas1	Katutie 1	Pori	28100	<input checked="" type="checkbox"/>									
<table border="1"> <thead> <tr> <th>Kontaktihenkilö</th> <th>Sähköposti</th> <th>Puhelinnumero</th> </tr> </thead> <tbody> <tr> <td>1 Yhteyshenkilö</td> <td></td> <td>0441234567</td> </tr> <tr> <td>Testi Teppo</td> <td>teppo@testi.com</td> <td>0441234567</td> </tr> </tbody> </table>					Kontaktihenkilö	Sähköposti	Puhelinnumero	1 Yhteyshenkilö		0441234567	Testi Teppo	teppo@testi.com	0441234567
Kontaktihenkilö	Sähköposti	Puhelinnumero											
1 Yhteyshenkilö		0441234567											
Testi Teppo	teppo@testi.com	0441234567											
Asiakas2	Katutie 2	Pori	28100	<input checked="" type="checkbox"/>									
Asiakas3	Katutie 3	Pori	28100	<input checked="" type="checkbox"/>									

« 1 »

© 2019 - Codecontrol + Suomi

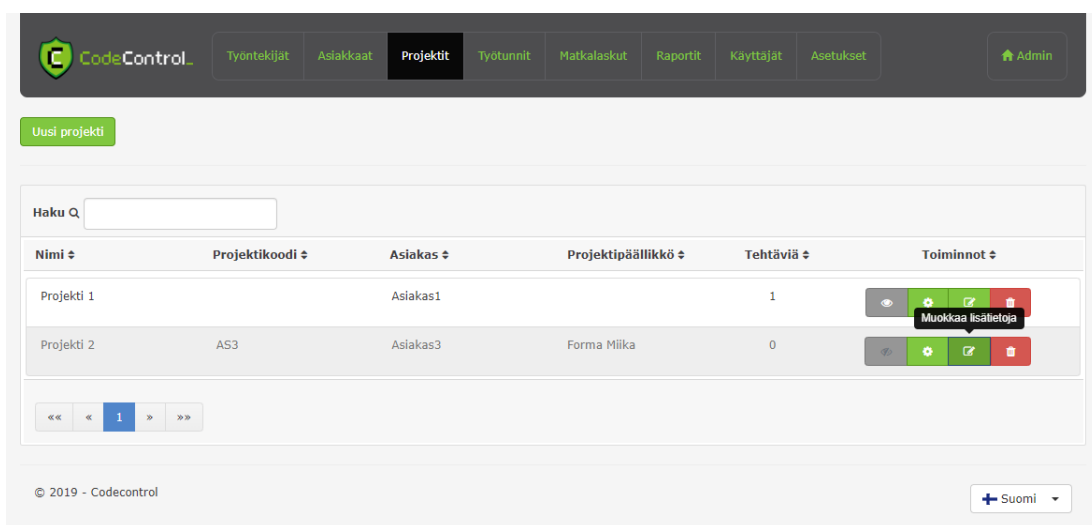
Kuva 13. Asiakkaat-näkymä.

Asiakkaita päästään lisäämään “Asiakkaat”-sivulla, jossa on lähes samanlainen näkymä kuin ”Työntekijät”-sivulla. Asiakkaita pääsee siis lisäämään, muokkaamaan, poistamaan sekä deaktivoimaan. Näiden lisäksi asiakkaille päästään kuitenkin lisäämään myös yhteyshenkilöitä (Kuva 14).



Kuva 14. Yhteyshenkilön lisäys asiakkaalle.

4.4 Projektit



Nimi	Projektkoodi	Asiakas	Projektipäällikkö	Tehtäviä	Toiminnot
Projekti 1		Asiakas1		1	[Muokkaa lisätietoja]
Projekti 2	AS3	Asiakas3	Forma Miika	0	[Toiminnot]

Kuva 15. Projektit-näkymä.

”Projektit”-näkymä seuraa samaa kaavaa ja näkymä näyttää lähes samanlaiselta kuin ”Työntekijät” ja ”Asiakkaat” -näkymät. Projektit-näkymällä näkyy kuitenkin yksi lisätoiminto, jonka kautta päästään lisäämään projektille tehtäviä sekä niiden tekijöitä (Kuva 15).

Muokkaa projektia [X]

Lisää tehtävä

Eka tehtävä Tehtäväkoodi Tehtävän kuvaus [Settings] [User] [Delete]

Työntekijä	Kuvaus	Yksikköhinta	Valuutta	Asiakkaalla
Forma Miika	Pori	10	Euro (€)	<input type="checkbox"/>
Forma Miika	Pori (Asiakkaalla)	20	Euro (€)	<input checked="" type="checkbox"/>

Uusi hinta

Toka tehtävä koodi Tehtävän kuvaus [Settings] [User] [Delete]

Sulje Tallenna

Kuva 16. Projektin tehtävien muokkaus.

Tehtäviä lisätessä niille tulee antaa nimi, kuvaus sekä tehtäväkoodi (jos sellainen on). Projektille voidaan lisätä useita eri tehtäviä ja yhdellä tehtävällä voi olla useita eri tekijöitä. Työntekijöitä tehtävälle lisätään painamalla ”Uusi hinta”-nappia ja valitsemalla työntekijä alavetovalikosta, jossa näkyy kaikki aktiiviset työntekijät (Kuva 16).

4.5 Työtunnit

4.5.1 Tuntikirjaukset

CodeControl Työntekijät Asiakkaat Projektit **Työtunnit** Matkalaskut Raportit Käyttäjät Asetukset Admin

Forma Miika ☰ Lähetä kirjaukset

lokakuu 2019

ma	ti	ke	to	pe	la	su
30	1 7h30m	2 7h30m	3 7h30m	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Pvm	Kohde	Kuvaus	Tunnit	Työn tyyppi	Tila
1.10.2019	Projekti 1 : Eka tehtävä	Kirjauksen kuvaus	7h 30m	Pori	Ei lähetetty
2.10.2019	Projekti 1 : Eka tehtävä	Keskiviikon kuvaus	7h 30m	Pori	Ei lähetetty
3.10.2019	Projekti 1 : Toka tehtävä	Torstain tunnit	7h 30m	Pori	Ei lähetetty

Valitse kaikki

1

Tehdyt työtunnit yhteensä: 22h 30m
Kirjattuja työtunteja yhteensä: 22h 30m
Laskutettavia työtunteja yhteensä: 22h 30m

Kuva 17. Tuntikirjaukset-näkymä.

“Tuntikirjaukset”-näkymä on sovelluksen yksi tärkeimmistä ominaisuuksista. Sen avulla kirjataan tehdyt tunnit. Tuntien kirjaaminen tapahtuu niin, että kalenterista valitaan päivämäärä ja napsautetaan hiiren kakkospainikkeella. Tämän jälkeen näkyviin tulee valikko, josta voi valita tuntikirjauksen tai lomamerkinän lisäämisen (Kuva 18).

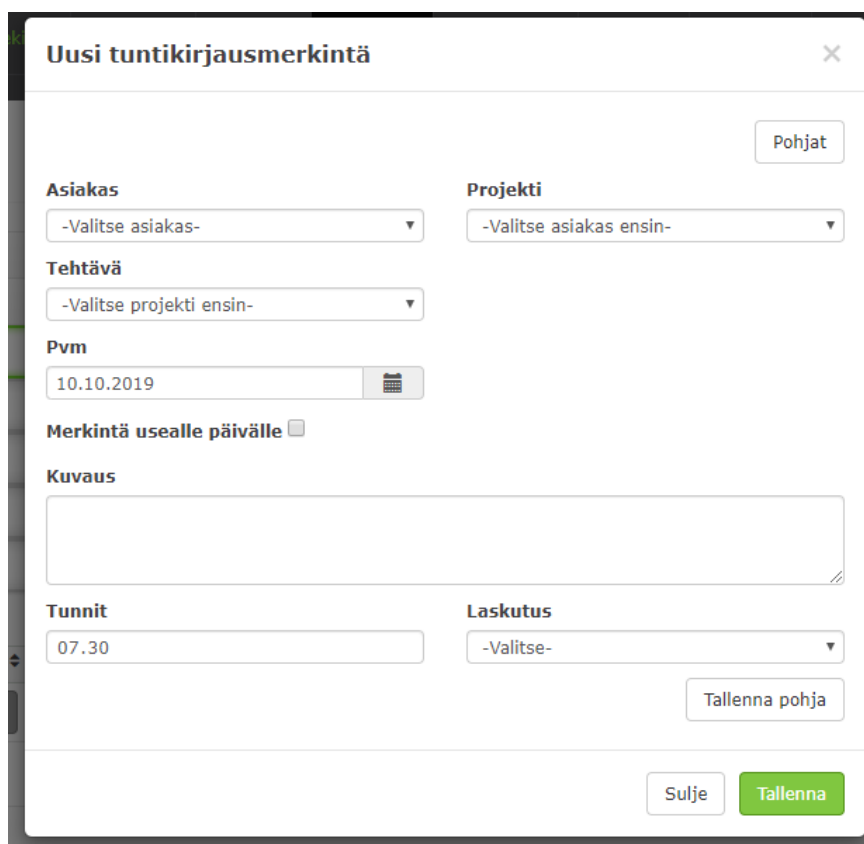
lokakuu 2019

ma	ti	ke	to	pe	la	su
30	1 7h30m	2 7h30m	3 7h30m	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Lisää merkintä päivälle
Lisää lomamerkintä päivälle

Kuva 18. Tuntikirjauksen tai lomamerkinän lisääminen.

Seuraavana vuorossa on tuntikirjauksen tietojen täyttäminen aukeavassa ikkunassa (Kuva 19). Ensimmäiseksi valitaan asiakas, jolle työ on tehty. Tämän jälkeen ohjelma hakee kaikki projektit, joita asiakkaalla on, ja joihin työntekijä on liitetty. Seuraavaksi valitaan projekti, joka hakee automaattisesti kaikki kyseisen projektin tehtävät, joihin työntekijä liitettiin kuvan 16 valikossa. Samalla kun valitaan tehtävä, haetaan kaikki laskutusvaihtoehdot, jotka kyseisen työntekijän sillä tehtävällä on. Laskutusvaihtoehdot tulevat kuvan 16 ”Kuvaus”-kentistä. Työntekijä voidaan liittää samaan tehtävään useampaan kertaan, jos halutaan käyttää eri hintaa, kun ollaan esimerkiksi asiakkaalla.



The screenshot shows a web form titled "Uusi tuntikirjausmerkintä" (New time recording entry). The form contains the following fields and controls:

- Asiakas** (Client): A dropdown menu with the placeholder text "-Valitse asiakas-".
- Projekti** (Project): A dropdown menu with the placeholder text "-Valitse asiakas ensin-".
- Tehtävä** (Task): A dropdown menu with the placeholder text "-Valitse projekti ensin-".
- Pvm** (Date): A date input field showing "10.10.2019" and a calendar icon.
- Merkintä usealle päivälle** (Entry for multiple days): A checkbox that is currently unchecked.
- Kuvaus** (Description): A large text area for entering details.
- Tunnit** (Hours): A text input field containing "07.30".
- Laskutus** (Billing): A dropdown menu with the placeholder text "-Valitse-".
- Buttons:** "Pohjat" (Basics) at the top right, "Tallenna pohja" (Save template) below the billing dropdown, "Sulje" (Close) and "Tallenna" (Save) at the bottom right.

Kuva 19. Uuden tuntikirjausmerkinnän lisäämisen.

”Kuvaus” ja ”Tunnit” -kentät ovat juurikin tuntikirjauksen kuvauksen ja tehtyjen tuntimäärien täyttämiseen. ”Tunnit”-kentässä on käytössä HTML5:n ”time”-tyyppi, jonka ansiosta esimerkiksi mobiililla, kenttää klikkaamalla aukeaa helppokäyttöinen ajan valitsin.

Joissain tapauksissa voi olla, että halutaan tallentaa sama kirjaus useille eri päiville. Sen sijaan, että se tarvitsisi täyttää joka päivälle uudelleen, voidaan valita ”Merkintä usealle päivälle”-valintaruutu, jonka jälkeen aukeaa toinen päivämäärän valinta.

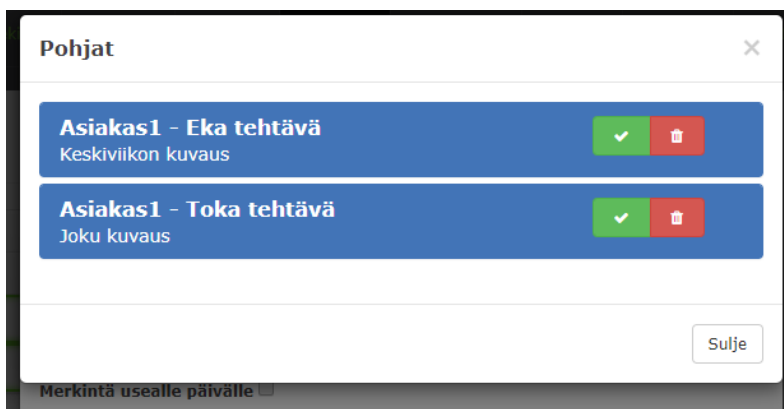
Päivämääriä säätämällä voidaan määrittää mille välille kyseinen merkintä tallennetaan. Esimerkiksi kuvan 20 merkintä tallennetaan kolmelle päivälle 9.10.2019 – 11.10.2019. Toteutuksessa on käytetty ”bootstrap-datepicker”-laajennusta.

The image shows a web form for creating a new time recording entry. The form is titled "Uusi tuntikirjausmerkintä". It contains several input fields and a date picker. The date picker is open, showing the month of October 2019. The dates 9, 10, and 11 are highlighted, indicating the selected period. The form also includes a "Tallenna pohja" button, a "Sulje" button, and a "Tallenna" button.

Kuva 20. Saman tuntikirjauksen merkitseminen useammalle päivälle.

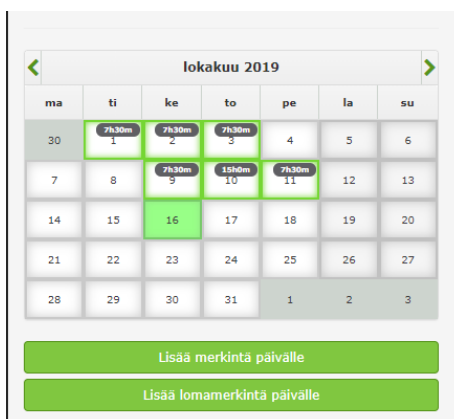
Kuvassa 20 näkyy myös painikkeet ”Tallenna pohja” sekä ”Pohjat”. Nämä ovat sitä varten, että ei tarvitse joka kerta täyttää kaikkia kenttiä, kun täyttää samaan tehtävään tuntikirjauksia. ”Tallenna pohja”-painike tallentaa kaiken muun paitsi päivämäärän. ”Pohjat”-painike avaa uuden dialogin vanhan päälle ja näyttää siinä tallennetut pohjat (Kuva 21). Vihreää painiketta painamalla voi valita kyseisen pohjan ja tämä lataa tuntikirjausmerkintä ikkunan pohjan tiedoilla, mutta aiemmin valitulla päivämäärällä.

Aikojen saatossa pohjia saattaa kertyä useita ja projektin valmistuttua pohjaa ei enää tarvita. Pohjan saakin poistettua painamalla punaista painiketta, jonka jälkeen ohjelma varmistaa vielä haluatko varmasti poistaa kyseisen pohjan. Tämän jälkeen pohjat lista latautuu uudelleen, jos tallennettuja pohjia on vielä jäljellä.



Kuva 21. Tuntikirjauksen tallennettuja pohjia.

Suurin osa asioista toimii mobiililaitteilla suoraan ilman, että tarvitsisi tehdä mitään muutoksia. Tukea hiiren kakkospainikkeelle mobiililaitteissa ei kuitenkaan suoranaisesti ole, joten kuvan 18 valikkoa ei niillä saada ilman lisäkoodia esille. Joillain mobiililaitteilla pitkään painaminen simuloi hiiren kakkospainiketta, mutta selkeyden vuoksi mobiililaitteille on lisätty painikkeet kalenterin alapuolelle (Kuva 22).



Kuva 22. Mobiililaitteen tuntikirjauspainikkeet.

Lomamerkinnän luominen tapahtuu lähes samalla tavalla kuin tuntikirjauksen lisääminen. Ainoana erona on hieman suppeampi määrä täytettäviä kohtia kuten näkyy kuvassa 23.

Luo lomamerkintä ✕

Loman tyyppi

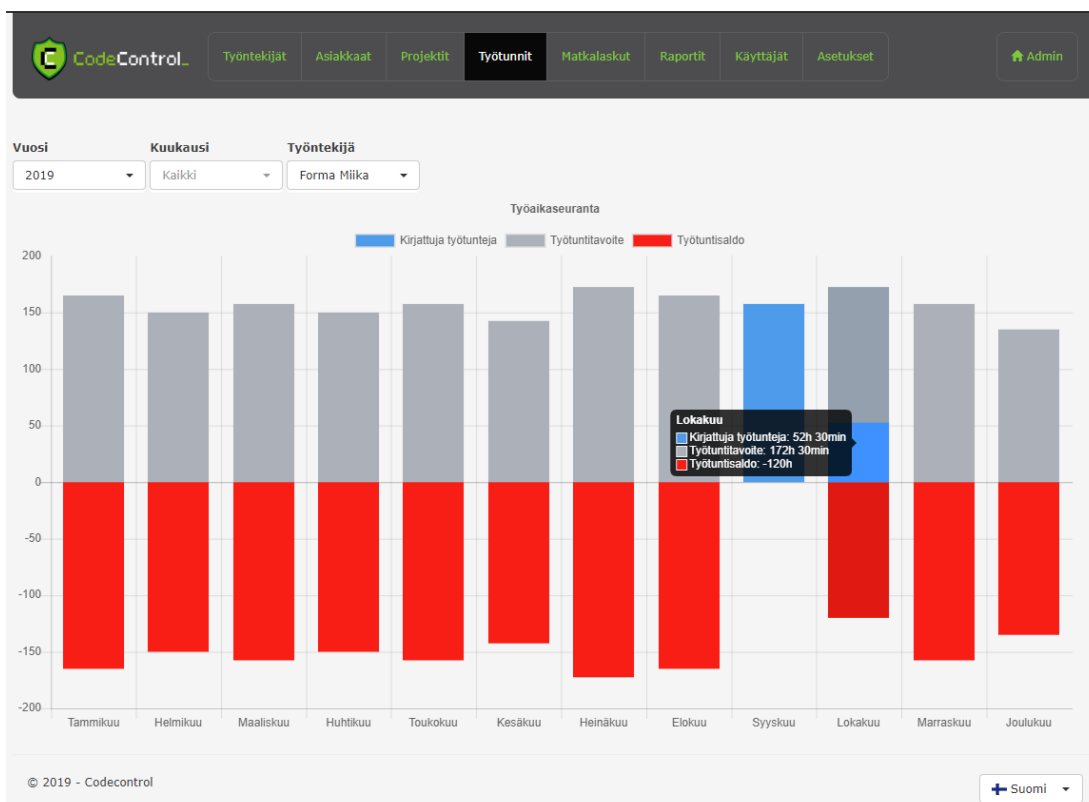
Pvm

Kuvaus

Tunnit

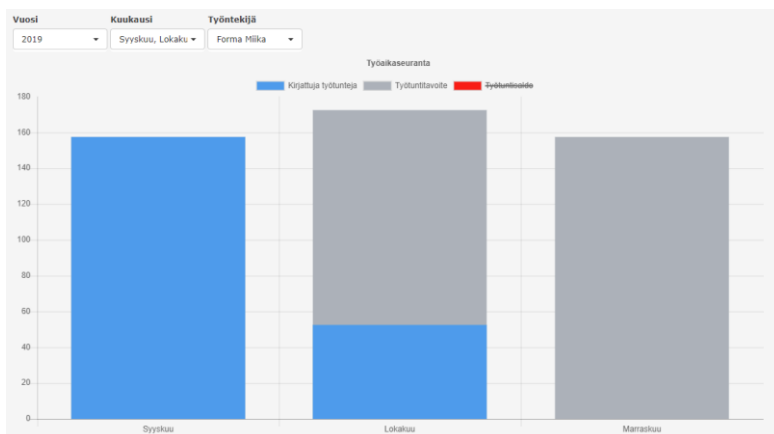
Kuva 23. Lomamerkinnän luominen.

4.5.2 Työaikaseuranta



Kuva 24. Työaikaseuranta-näkymä.

”Työaikaseuranta”-näkyssä näkee kunkin kuukauden tuntikertymän sekä kuinka paljon työtuntitavoite kyseiselle kuukaudelle oli. Grafiikan näyttämiseen on käytetty ”Chart.js”-nimistä kirjastoa. Grafiikkaa voi muuttaa niin, että minkä tahansa ”Kirjattuja työtunteja”, ”Työtuntitavoite” tai ”Työtuntisaldo” kentistä voi piilottaa. On myös mahdollista valita, mitkä kuukaudet haluat nähdä, kuten kuvassa 25 on demonstroitu. Työaikaseurannan laskuihin käytetyt arvot täytetään asetukset sivulta kuvassa 33.



Kuva 25. Työaikaseuranta kolmella kuukaudella ja ilman työtuntisaldoa.

4.6 Matkalaskut

The screenshot shows the CodeControl application interface. The navigation menu includes: Työntekijät, Asiakkaat, Projektit, Työtunnit, **Matkalaskut**, Raportit, Käyttäjät, Asetukset, and Admin. The main content area displays the 'Matkalaskut' section for October 2019. The table below shows the details of travel expenses.

Aloitus pvm	Päätymis pvm	Matkustus aika	Projekti	Matkan pituus	Päiväraha kpl	Päiväraha	Km á	Yhteensä	Reitti/Kuvaus	Tila
1.10.2019	1.10.2019	2h	Projekti 1	250	0	0,00	132,5	132,5	Route description	Ei lähetetty

Below the table, there is a 'Valitse kaikki' button. The interface also shows a section for 'Muut matkakulut' (Other travel expenses) with a table:

Pvm	Projekti	Kuvaus	Tila
1.10.2019	Projekti 1	Joku kuvaus	Ei lähetetty

At the bottom of the page, there is a copyright notice: © 2019 - Codecontrol and a language selector: + Suomi.

Kuva 26. Matkalaskut-näkymä.

“Matkalaskut”-sivulla täytetään pääasiassa omalla autolla työn puolesta tehtyjen matkojen kirjauksia, joista maksetaan kilometrikorvauksia. Sivulla on myös mahdollista täyttää muita matkakuluja, joita on saattanut tulla.

Muokkaa matkaa

Asiakas
Asiakas1

Aloituspvm
1.10.2019 09.00

Vuosi
2019

Maa
Suomi

Kulkuneuvo
Oma auto (0,43)

Reitti/Kuvaus
Route description

Projekti
Projekti 1

Päättyminen pvm
1.10.2019 15.00

Laskutettava

Liitä tiedosto
Valitse tiedosto Ei valittua tiedostoa

Mittarilukema alkaessa
0

Mittarilukema päättyessä
250

Matkustusaika
02.00

Puolipäiväraha
0

Kilometrikorvauksen korotukset
Peräkärä (0,07), Koira (0,03)

Päiväraha kpl	Päiväraha	Kilometrikorvaus yht.	Matkan pituus (km)
0.00	0.00	132.50	250

Poista
Sulje
Tallenna

Kuva 27. Matkalaskun muokkaus/lisäys.

Matkalaskun lisäämisessä on mahdollista täyttää monia kenttiä. Kuvassa 27 on valittuna oma auto, joten täytettäväksi tulee kenttiä, joiden mukaan lasketaan kilometrikorvaukset. Sen sijaan esimerkiksi junalla kulkiessa näitä kenttiä ei täytetä (Kuva 28). Kenttien näkymistä kontrolloidaan JavaScript-koodilla.

Luo uusi matkakulu

Asiakas
Asiakas1

Aloituspvm
1.10.2019 09.00

Vuosi
2019

Maa
Suomi

Kulkuneuvo
Juna (0,00)

Reitti/Kuvaus

Projekti
Projekti 1

Päättyminen pvm
1.10.2019 15.00

Laskutettava

Liitä tiedosto
Valitse tiedosto Ei valittua tiedostoa

Matkustusaika
00.00

Puolipäiväraha
0

Päiväraha kpl	Päiväraha	Kilometrikorvaus yht.	Matkan pituus (km)
0.00	0.00	0.00	0

Sulje
Tallenna

Kuva 28. Matkalaskun lisäys junalla kulkiessa.

4.7 Raportit

”Raportit”-sivulla päästään katsomaan ns. yhteenvetoja tuntikirjauksista, matkalaskuista sekä muista matkakuluista (Kuva 29). ”Tuntikirjaukset”-kohdassa olevia suodattimia käyttämällä saadaan näkyviin vain halutut kirjaukset. Suodattimet koskevat myös matkalaskuja sekä muita matkakuluja, vaikka ne ovatkin tuntikirjaus otsikon alla.

Taulujen päivittäminen toimii Ajax-kutsuilla niin, että jokaisesta taulusta lähtee erillinen pyyntö palvelimelle. Palvelin palauttaa takaisin HTML-koodia, joka sisältää uudet tiedot. Tämän jälkeen JavaScript-koodilla vaihdetaan sivulla olevan taulun koodi uuteen. Näin saadaan taulujen tiedot päivitettyä ilman, että koko sivua tarvitsee ladata uudelleen.

Tuntikirjaukset											
Asiakas	Työntekijä	Projekti	Aloitus pvm	Päätymis pvm							
Kaikki	Kaikki	Kaikki	1.10.2019		31.10.2019						
Pvm	Kohde	Kuvaus	Tunnit	Työntekijä	Työn tyyppi	Tila					
1.10.2019	Projekti 1 : Eka tehtävä	Kirjauksen kuvaus	7h 30min	Forma Miika	Pori	Ei lähetetty					
2.10.2019	Projekti 1 : Eka tehtävä	Keskiviikon kuvaus	7h 30min	Forma Miika	Pori	Ei lähetetty					
3.10.2019	Projekti 1 : Toka tehtävä	Torstain tunnit	7h 30min	Forma Miika	Pori	Ei lähetetty					
9.10.2019	Projekti 1 : Eka tehtävä	Tuntikirjauksen kuvaus	7h 30min	Forma Miika	Pori	Ei lähetetty					
10.10.2019	Projekti 1 : Eka tehtävä	Testi	7h 30min	Forma Miika	Pori	Ei lähetetty					
10.10.2019	Projekti 1 : Eka tehtävä	Tuntikirjauksen kuvaus	7h 30min	Forma Miika	Pori	Ei lähetetty					
11.10.2019	Projekti 1 : Eka tehtävä	Tuntikirjauksen kuvaus	7h 30min	Forma Miika	Pori	Ei lähetetty					
Yhteensä:			52h 30m								
« 1 »											
Matkalaskut											
Aloitus pvm	Päätymis pvm	Työntekijä	Matkustusaika	Projekti	Matkan pituus	Päiväraha kpl	Päiväraha	Km á	Yhteensä	Reitti/Kuvaus	Tila
1.10.2019	1.10.2019	Forma Miika	2h	Projekti 1	250	0	0,00	132,5	132,5	Route description	Ei lähetetty
Yhteensä:			2h 0m	250	0	0,00	132,5	132,5			
« 1 »											
Muut matkakulut											
Pvm	Työntekijä	Projekti	Määrä	Kuvaus	Tila						
1.10.2019	Forma Miika	Projekti 1	25	Joku kuvaus	Ei lähetetty						
Yhteensä:			25								
« 1 »											

Kuva 29. ”Raportit”-sivun taulut.

Sivun yläreunassa on myös painikkeet, joiden avulla saadaan raportteja ulos Excel-tiedostoina. Esimerkki matkalasku ja muu matkakulu raportin luonnista on kuvassa 30, jossa siis voidaan valita mitä kenttiä raporttiin tulee sekä millä nimellä se tallennetaan.

Raportin asetukset

Tiedostonimi
Matkalaskut_10-19

Valitse haluamasi matkalasku sarakkeet
Kaikki


Valitse haluamasi muu matkakulu sarakkeet
Kaikki

Valitse kaikki Poista kaikki


Pvm
Työntekijä
Projekti
Laskutettava
Kausi
Määrä
Kuvaus

Kuva 30. Raportin luonti.

Esimerkki luodusta tuntikirjaus raportista löytyy kuvasta 31 ja matkalasku sekä muu matkakulu raportista kuvasta 32. Raportit generoidaan EPPlus-kirjaston avulla.

	A	B	C	D	E	F	G
1		CodeControl					8.10.2019
2							
3							
4							
5							
6							
7							
8							
9							
10	Pvm	Kohde	Kuvaus	Tunnit	Työntekijä	Yksikköhinta	Laskutus
11	1.10.2019	Projekti 1 : Eka tehtävä	Kirjauksen kuvaus	7h 30min	Forma Miika	10,00 €	75,00 €
12	2.10.2019	Projekti 1 : Eka tehtävä	Keskiviikon kuvaus	7h 30min	Forma Miika	10,00 €	75,00 €
13	3.10.2019	Projekti 1 : Toka tehtävä	Torstain tunnit	7h 30min	Forma Miika	20,00 €	150,00 €
14	9.10.2019	Projekti 1 : Eka tehtävä	Tuntikirjauksen kuvaus	7h 30min	Forma Miika	10,00 €	75,00 €
15	11.10.2019	Projekti 1 : Eka tehtävä	Tuntikirjauksen kuvaus	7h 30min	Forma Miika	10,00 €	75,00 €
16							
17	Yhteensä	52h 30min					

Kuva 31. Tuntikirjaus-raportin esimerkki.







	A	B	C	D	E	F	G	H	I	J	K	L
1		CodeControl										9.10.2019
2												
3												
4												
5												
6												
7												
8												
9												
10	Aloitus pvm	Päätymis pvm	Matkustusaika	Työntekijä	Projekti	Matkan pituus	Päiväraha kpl	Päiväraha	Km á	Laskutettava	Yhteensä	Reitti/Kuvaus
11	1.10.2019	1.10.2019	2h	Forma Miika	Projekti 1	250	0	0,00	132,5	Ei	132,5	Route description
12												
13												
14												
15	Pvm	Työntekijä	Projekti	Laskutettava	Kausi	Määrä	Kuvaus					
16	1.10.2019	Forma Miika	Projekti 1	Ei	Lokakuu	25	Joku kuvaus					
17												
18	MATKALASKU	132.50 €										
19	MUUT MATKAKULUT	25.00 €										
20	Yhteensä	157.50 €										

Kuva 32. Matkalasku/muu matkakulu -raportin esimerkki.

4.8 Käyttäjät

Luo käyttäjä

Haku Q

Käyttäjänimi	Liitetty	Sähköposti	Toiminnot
Admin	-		 
Työntekijä	Miika Forma	miika.forma@testi.com	 
Valvoja	Miika Forma	miika.forma@testi.com	 

«<< < 1 > >>»

© 2019 - Codecontrol + Suomi

Kuva 33. Käyttäjät-näkymä.

”Käyttäjät”-näkymässä päästään lisäämään, muokkaamaan ja poistamaan tunnuksia, joilla järjestelmään pääsee kirjautumaan. Roolivaihtoehtoina ovat ”Admin”, ”Valvoja” sekä ”Työntekijä”. Admin-roolilla pääsee käyttämään järjestelmän kaikkia ominaisuuksia.

Valvoja-roolia on rajoitettu hieman, joten sillä ei pääse esimerkiksi katsomaan toisen työntekijän kirjauksia tai muokkaamaan käyttäjiä tai työntekijöitä. Sillä pääsee kuitenkin muokkaamaan ohjelman asetuksia sekä esimerkiksi lisäämään projekteja.

Työntekijä-rooli on kaikista rajoitetuin, joten sillä ei pääse muokkaamaan mitään ohjelman asetuksia. Sillä pääsee kuitenkin katsomaan työntekijöiden yhteystiedot sekä sillä pääsee lisäämään tunti- ja lomakirjauksia sekä matkalaskuja sekä muita matkaku-
luja. Työntekijä pääsee myös tekemään Excel-raportteja omista kirjauksistaan.

4.9 Asetukset

The screenshot shows the 'Asetukset' (Settings) page in the CodeControl system. The top navigation bar includes 'Työntekijät', 'Asiakkaat', 'Projektit', 'Työtunnit', 'Matkalaskut', 'Raportit', 'Käyttäjät', and 'Asetukset' (highlighted). There is also an 'Admin' button.

The main content area is divided into two sections:

- Valuutat**: A section for managing currencies. It shows a single entry for 'Euro (€)' with edit and delete icons.
- Lomatyytit**: A section for managing leave types. It contains a table with the following data:

Yliytövapaa	Nimi	Toiminnot
<input checked="" type="checkbox"/>	Ylityö	[edit] [delete]
<input type="checkbox"/>	Vuosiloma	[edit] [delete]

At the bottom, there is a copyright notice '© 2019 - Codecontrol' and a language selector set to 'Suomi'.

Kuva 34. Yleiset asetukset -näkömä.

”Asetukset”-sivulla päästään muokkaamaan käytössä olevia valuuttoja sekä lomatyypejä. Valuutat, jotka lisätään asetuksissa, ovat valittavissa projektin tehtävän muokkauksessa (Kuva 16). Lomatyytit ovat niitä vaihtoehtoja, jotka näkyvät lomamerkin- töjä kirjatessa, kuten kuvassa 23.

The screenshot shows the 'Asetukset' (Settings) page in the CodeControl system, specifically the 'Matkalaskuasetukset' (Travel Expense Settings) section. The top navigation bar is the same as in the previous screenshot.

The main content area includes:

- A year selector: '+ Lisää vuosi' and buttons for 2015, 2016, 2017, 2018, and 2019.
- Lisämatkustajat**: A section for additional travelers. It shows a table with the following data:

Yksikköhinta			Toiminnot
0,03			[edit] [delete]
- Kulkuneuvot**: A section for transport modes. It shows a table with the following data:

Kulkuneuvo	Yksikköhinta	Matkakorvauksen korotukset	Muokkaa matkakorvauksia	Globaalit toiminnot
Oma auto	0,43	4 kpl	[edit]	[eye] [edit] [delete]
Bussi		0 kpl	[edit]	[eye] [edit] [delete]
Juna	0,00	0 kpl	[edit]	[eye] [edit] [delete]
- Päivärahämäärät**: A section for daily allowance amounts. It shows a table with the following data:

Maa	Kokopäiväraha	Osapäiväraha	Aktiivinen	Toiminnot
Suomi	42,00	19,00	<input checked="" type="checkbox"/>	[eye] [edit] [delete]

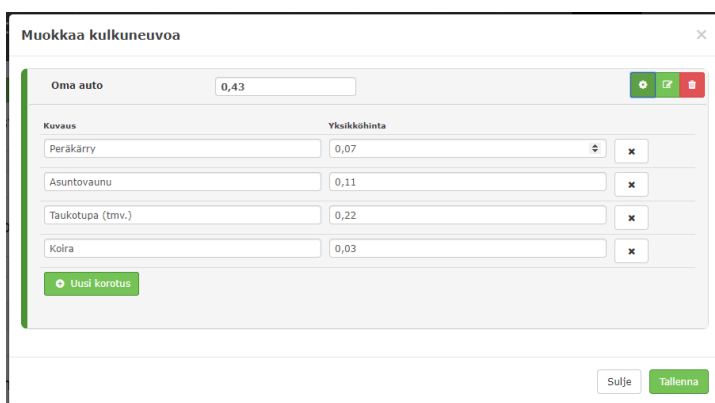
At the bottom, there is a copyright notice '© 2019 - Codecontrol' and a language selector set to 'Suomi'.

Kuva 35. Matkalaskuasetukset-näkömä.

Matkalaskujen asetuksissa asetetaan arvot, joiden perusteella lasketaan mm. kilometri- korvaukset ja päivärahat. ”Lisää vuosi”-painikkeen takaa aukeaa ikkuna, josta

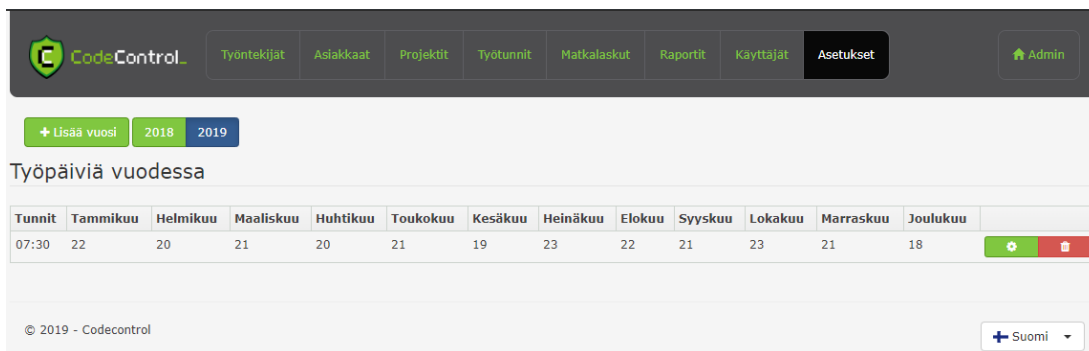
päästään lisäämään uusi vuosi ja siihen liittyvät ”Kokopäiväraha” sekä ”Osapäiväraha” -arvot. Sen jälkeen sivulle latautuu lisätyn vuoden asetukset, joista pääsee muokkaamaan lisämatkustajien, kulkuneuvojen sekä päivärahamäärien asetuksia.

Kulkuneuvot ovat kaikille vuosille samoja, mutta niitä voi piilottaa näkyvistä vuosikohtaisesti. Uuden vuoden lisäyksen jälkeen kaikki aikaisempien vuosien kulkuneuvot ovatkin siis poissa käytöstä, kunnes niille annetaan oikeat yksikköhinnat. Jokaiselle kulkuneuvolle voi myös lisätä korotuksia, jos sellaisille on tarvetta (Kuva 36). Lisämatkustajien korotus on asetettavissa erikseen, koska sen yksikköhinta vaihtelee lisämatkustajien määrästä, toisin kuin muiden korvauksien.



Kuva 36. Kilometrikorvauksen asetukset sekä korotuksien lisäys.

Päivärahamääriin voidaan lisätä myös muiden maiden päivärahamääriä, jos sellaisille on tarvetta. Suomi on päivärahamäärissä pakollinen ja sitä poistaessa varoitetaan, että poistossa käytöstä poistetaan koko vuosi, eikä vain päivärahamäärää.



Tunnit	Tammikuu	Helmikuu	Maaliskuu	Huhtikuu	Toukokuu	Kesäkuu	Heinäkuu	Elokuu	Syyskuu	Lokakuu	Marraskuu	Joulukuu
07:30	22	20	21	20	21	19	23	22	21	23	21	18

Kuva 37. Työpäiväasetukset-näkymä.

Työpäiväasetuksien arvoja käytetään työaikaseurannan graafin näyttämiseksi, jota näytettiin kuvassa 24.

5 LOGIEN TEKEMINEN

5.1 Loki käytön seurantaan

Turvallisuussyistä sovellukseen on lisätty loki käytön seurannasta, jotta tarvittaessa päästään katsomaan, kuka on tehnyt jonkin asian. Eli pystytään esimerkiksi tunnistamaan, mistä osoitteesta ollaan pyritty kirjautumaan sisään arvailemalla tunnuksia. Tai pystytään katsomaan, kuka on esimerkiksi poistanut projektin.

Auditointi on toteutettu niin, että sille on tehty oma attribuutti, joka tallentaa tiedot tietokantaan (Kuva 38). Tietoja ei kuitenkaan tallenneta kaikista ohjaimen metodeista, vaan vain niistä, jotka on koristeltu ”Audit”-attribuutilla (Kuva 39).

```
public class AuditAttribute : ActionFilterAttribute
{
    1 reference | 0 changes | 0 authors, 0 changes
    public bool IncludeForm { get; set; }
    15 references | 0 changes | 0 authors, 0 changes
    public string Action { get; set; }

    0 references | 0 changes | 0 authors, 0 changes
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var fallbackAction = filterContext?.ActionDescriptor?.ActionName ?? "Unknown";

        var request = filterContext.HttpContext.Request;
        var newAudit = new Audit() {
            UserName = (request.IsAuthenticated) ? filterContext.HttpContext.User.Identity.Name : "Anonymous",
            IPAddress = request.ServerVariables["HTTP_X_FORWARDED_FOR"] ?? request.UserHostAddress,
            AreaAccessed = request.RawUrl,
            Timestamp = DateTime.UtcNow,
            Action = Action ?? fallbackAction
        };
    }
}
```

Kuva 38. Osa Audit-attribuutin koodista.

```
//
// POST: /Account/Login
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
[Audit]
3 references | 1 author, 1 change
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
}
```

Kuva 39. Audit-koristelu, jonka seurauksena kirjautumisia tallennetaan lokiin.

5.2 Virheloki

Virheiden kirjaaminen on tärkeää, jotta niihin johtaneet syyt olisi helpompi selvittää. Virhe voi olla myös sellainen, että sitä on hankala toistaa, joten virheen kirjaaminen voi olla ainoa tapa päästä syyn jäljille.

Virheiden kirjaaminen tässä sovelluksessa on hoidettu ELMAH-paketin avulla. ELMAH kirjaa virheet XML-tiedostoihin, joita pääsee katsomaan joko levynkulmalta tai sovelluksen kautta osoitteesta ”~/elmah.axd” (Kuva 40).

The controller for path '/eiloydy' was not found or does not implement IController.

```

System.Web.HttpException (0x80004005): The controller for path '/eiloydy' was not found or does not implement IController.
kohteessa System.Web.Mvc.DefaultControllerFactory.GetControllerInstance(RequestContext requestContext, Type controllerType)
kohteessa System.Web.Mvc.DefaultControllerFactory.CreateController(RequestContext requestContext, String controllerName)
kohteessa System.Web.Mvc.MvcHandler.ProcessRequestInit(HttpContextBase httpContext, IController controller, IControllerFactory factory)
kohteessa System.Web.Mvc.MvcHandler.BeginProcessRequest(HttpContextBase httpContext, AsyncCallback callback, Object state)
kohteessa System.Web.Mvc.MvcHandler.BeginProcessRequest(HttpContext httpContext, AsyncCallback callback, Object state)
kohteessa System.Web.Mvc.MvcHandler.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context, AsyncCallback cb, Object extraData)
kohteessa System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()
kohteessa System.Web.HttpApplication.ExecuteStepImpl(IExecutionStep step)
kohteessa System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously)

```

Kuva 40. ELMAH-kirjaston kirjaama ja näyttämä virhe.

6 TESTAUS

Sovellukseen on kirjoitettu testejä, jotta voidaan tarkistaa ohjelman toiminta automaattisesti koodimuutoksien jälkeen. Koko sovelluksen läpikäyminen manuaalisesti esimerkiksi uuden ominaisuuden lisäämisen jälkeen olisi liian työlästä. Testien avulla voidaan varmistaa, että vanha koodi toimii ilman, että kaikkia toimintoja tarvitsisi testata käsin.

Sovellukseen on kirjoitettu yli 400 yksikkötestiä, joiden avulla testataan ohjelman oikein toimimista. Niillä testataan myös, että ohjelma ei kaadu, vaikka metodeille annettaisiin niille kuulumattomia arvoja.

```

18 using System.Web.Mvc;
19 using System.Web.Routing;
20
21 namespace CcTracker.Tests
22 {
23     [TestClass]
24     public class HelperTests
25     {
26         #region UserHelpers
27         [TestMethod]
28         [TestCategory("Helpers_UserHelpers")]
29         public void UserHelpers_LocalizeRole()
30         {
31             // Assign
32
33             // Act
34             var adminresult = UserHelper.LocalizeRole(Roles.Admin);
35             var employeeResult = UserHelper.LocalizeRole(Roles.Employee);
36             var clientResult = UserHelper.LocalizeRole(Roles.Client);
37             var supervisorResult = UserHelper.LocalizeRole(Roles.Manager);
38             var emptyResult = UserHelper.LocalizeRole("FooBar");
39
40             // Assert
41             // Check Admin
42             Assert.AreEqual(Resources.Admin, adminresult);
43             // Check Employee
44             Assert.AreEqual(Resources.Employee, employeeResult);
45             // Check Client
46             Assert.AreEqual(Resources.Client, clientResult);
47             // Check Manager
48             Assert.AreEqual(Resources.Manager, supervisorResult);
49             // Check Empty
50             Assert.AreEqual(string.Empty, emptyResult);
51         }
52
53         [TestMethod]
54         [TestCategory("Helpers_UserHelpers")]
55         public void UserHelpers_UserExists()
56         {
57             // Assign
58
59             // Act
60
61             // Assert
62         }
63     }
64 }

```

Kuva 41. Sovellukseen kirjoitettuja testejä.

Kirjoitetut yksikkötestit testaavat kuitenkin vain metodien toimintaa. Käyttöliittymän testaamiseen on kirjoitettu testejä Robot Frameworkin avulla. Näiden testien avulla sovelluksen toiminta tulee testattua käyttäjän syötteestä lähtien palvelimelle ja takaisin. Muutamia testitapauksia on kirjoitettuna kuvassa 7 ja kuvassa 42 näkyy, kuinka itse testauskoodi on kirjoitettu.

```

CcTracker_common_keywords.robot
1  *** Settings ***
2  Library Selenium2Library
3
4  *** Variables ***
5  ${HOMEPAGE} http://localhost:1848
6  ${BROWSER} edge
7
8  *** Keywords ***
9  Go to loginpage
10 Open Browser ${HOMEPAGE} ${BROWSER}
11 Wait Until Page Contains Käyttäjänimi
12 Wait Until Page Contains Salasana
13
14 User logins "${USERNAME}" "${PASSWORD}"
15 Input Text id=UserName ${USERNAME}
16 Input Text id=Password ${PASSWORD}
17 Submit Form
18
19 User logout "${USERNAME}"
20 Click Link xpath=//*[contains(text(), '${USERNAME}')]
21 Click Link xpath=//*[contains(text(), 'Kirjaudu ulos')]
22 Wait Until Page Contains Käyttäjänimi
23 Wait Until Page Contains Salasana
24
25 Work page user selection is shown
26 Wait Until Page Contains Valitse käyttäjä
27
28 Invalid login is shown
29 Wait Until Page Contains Invalid login attempt.
30
31 Empty error is shown
32 Wait Until Page Contains Kenttä Käyttäjänimi on pakollinen.
33 Wait Until Page Contains Kenttä Salasana on pakollinen.

```

Kuva 42. Robot Framework -testit.

7 POHDINTA

Sovelluksen tekeminen oli mielenkiintoinen ja paljon opettava projekti. Ohjelmointia olen harrastanut jo useita vuosia, mutta lähes kaikki sovellukset ovat tulleet omaan käyttöön. Nyt edessä oli kuitenkin sellaisen sovelluksen tekeminen, joka tulee myös muiden käyttöön, ja jonka tulee tukea laitteita, joita itse en käytä.

Tekemisessä käytettiin useaa minulle entuudestaan tuntematonta tekniikkaa. ASP.NET MVC -sovelluksia olin jo aikaisemmin tehnyt, mutta tämän projektin aikana tuli mm. sähköpostien lähettäminen sekä testien tekeminen täysin uutena asiana. Näiden opetteleminen oli kuitenkin hyvinkin mielenkiintoista ja myös tulevaisuuden kannalta tärkeää.

Projektin aikana uusien tekniikoiden käyttö ei suinkaan ollut ainoa asia, jonka seurauksen ohjelmointitaidot kehittyivät. Käytössä olleiden koodikatselmointien seurauksena huomasin sen, että koodin laatu parani huomattavasti. Omissa projekteissa kukaan muu ei tarkasta koodia, joten tällaista ei tapahdu.

Haasteitakin sovelluksen tekemisessä oli mm. siinä, kun tuli yllättäviä ongelmia laitteilla, joita itselläni ei ollut käytössä. Esimerkiksi omalla Android-puhelimellani kuvan 16 kontekstivalikon sai auki pitkällä painalluksella. Tämä ei kuitenkaan toiminut iOS-mobiililaitteilla. Ongelma päästiin lopulta kiertämään lisäämällä mobiililaitteille eriliset nappulat kalenterin alapuolelle. Myös testien tekemisessä oli välillä hyvin haastavaa saada simuloitua tarvittavaa dataa.

Kaiken kaikkiaan projektia tehdessä opin paljon uusia tekniikoita ja koodin laatu parani huomattavasti.

LÄHTEET

Caliman, D. 2015. To user or not to use Bootstrap Framework? Viitattu: 27.09.2019. <https://blog.creative-tim.com/web-design/use-not-use-bootstrap-framework/>

Goel, A. 2019. MariaDB vs MySQL: [2019] Everything You Need to Know. Viitattu: 18.09.2019. <https://hackr.io/blog/mariadb-vs-mysql>

Harvest www-sivut. 2019. Why is time tracking important? Viitattu: 14.10.2019. <https://www.getharvest.com/resources/why-is-time-tracking-important>

Johansson, L. 2019. Part 1: RabbitMQ for beginners – What is RabbitMQ? Viitattu: 27.09.2019. <https://www.cloudamqp.com/blog/2015-05-18-part-1-rabbitmq-for-beginners-what-is-rabbitmq.html>

jQuery www-sivut. 2019. What is jQuery. Viitattu: 27.09.2019. <https://jquery.com/>

Kantor, I. 2019. JavaScript Fundamentals. Viitattu: 10.09.2019. <https://javascript.info/hello-world>

Microsoft www-sivut. 2019a. ASP.NET MVC Pattern. Viitattu: 20.08.2019. <https://dotnet.microsoft.com/apps/aspnet/mvc>

Microsoft www-sivut. 2019b. Introduction to the C# Language and the .NET Framework. Viitattu: 24.08.2019. <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

Microsoft www-sivut. 2019c. Hello World -- Your first program (C# Programming Guide). Viitattu: 24.08.2019. <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/hello-world-your-first-program>

Microsoft www-sivut. 2019d. Introduction to ASP.NET Identity. Viitattu: 28.08.2019. <https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity>

Mills, C. 2019. JavaScript. Viitattu: 10.09.2019. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Mitchell, S. 2009. Logging Error Details with ELMAH (C#). Viitattu: 28.08.2019. <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/older-versions-getting-started/deploying-web-site-projects/logging-error-details-with-elmah-cs>

Morris, S. 2018. Tech 101: The Ultimate Guide to CSS. Viitattu: 11.09.2019. <https://skillcrush.com/2012/04/03/css/>

Musso, E. 2016. Entity Framework Introduction Using C# - Part One. Viitattu: 03.09.2019. <https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/>

MySQLTUTORIAL www-sivut. 2019. What Is MySQL. Viitattu: 18.09.2019. <http://www.mysqltutorial.org/what-is-mysql/>

RabbitMQ [www-sivut](http://www.rabbitmq.com/). 2019. RabbitMQ is the most widely deployed open source message broker. Viitattu: 27.09.2019. <https://www.rabbitmq.com/>

RobotFramework [www-sivut](http://www.robotframework.org/). 2019. RobotFramework/. Viitattu: 27.09.2019. <https://robotframework.org/>

Saldanha, L. 2014. Untangling Concepts: Unit Tests vs Acceptance Tests. Viitattu: 27.09.2019. <https://dzone.com/articles/untangling-concepts-unit-tests>

Seravo [www-sivut](http://www.seravo.fi/). 2015. 10 reasons to migrate to MariaDB (if still using MySQL). Viitattu: 18.09.2019. <https://seravo.fi/2015/10-reasons-to-migrate-to-mariadb-if-still-using-mysql>

Shay, T. 2018. MariaDB vs MySQL – Comparing MySQL 8.0 with MariaDB 10.3. Viitattu: 18.09.2019. <https://www.eversql.com/mariadb-vs-mysql/>

SphinxKnight. 2019. CSS: Cascading Style Sheet. Viitattu: 11.09.2019. <https://developer.mozilla.org/en-US/docs/Web/CSS>

Tutorialspoint [www-sivut](http://www.tutorialspoint.com/). 2019a. ASP.NET MVC – Data Model. Viitattu: 24.08.2019. https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_data_model.htm

Tutorialspoint [www-sivut](http://www.tutorialspoint.com/). 2019b. ASP.NET MVC – Controllers. Viitattu: 24.08.2019. https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_controllers.htm

Tutorialspoint [www-sivut](http://www.tutorialspoint.com/). 2019c. ASP.NET MVC – Views. Viitattu: 24.08.2019. https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_views.htm

Tyosuojelu.fi [www-sivut](http://www.tyosuojelu.fi/). 2019. Työaika. Viitattu: 14.10.2019. <https://www.tyosuojelu.fi/tyosuhde/tyo aika>

wbamberg. 2019. HTML: Hypertext Markup Language. Viitattu: 10.09.2019. <https://developer.mozilla.org/en-US/docs/Web/HTML>