



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Nico Kauppila

# CRM-järjestelmä

Case Pilvilampi Software Oy

Liiketalous

2018

## TIIVISTELMÄ

Tekijä	Nico Kauppila
Opinnäytetyön nimi	CRM-järjestelmä
Vuosi	2018
Kieli	suomi
Sivumäärä	28
Ohjaaja	Sirkka Hellman

---

Tämä opinnäytetyö käsittelee Microsoft O365 järjestelmän päälle rakennettua asiakkuudenhallinta eli CRM-ratkaisua. Ohjelmoinnissa käytetään AngularJS -kirjastoja, sekä Bootstrap HTML-pohjaa. Työ on tehty toimeksiantona Pilvilampi Software Oy:lle.

Opinnäytetyön tavoitteena oli tehdä järjestelmä, jolla ylläpidetään myyntiä, sopimuksia, projekteja sekä Helpdesk-palvelua. Järjestelmän avulla yrityksen tärkeät tiedot tulevat olemaan yhdessä paikassa ja työskentelystä tulee helpompaa, koska kaikki tiedot ovat nopeasti kaikkien käytettävissä.

Yrityksen käytössä on ollut aiemmin ProjectWeb niminen samaan tarkoitukseen tarkoitettu järjestelmä, mutta tavoitteena oli sen monimutkaisuuden ja muokattavuuden takia luoda täysin oman järjestelmän, jota olisi mahdollista päivittää ja muokata yrityksen muuttuvien tarpeiden mukaan.

Työn tuloksena syntyi järjestelmä, joka on edeltäjänsä parempi sekä käytettävyydeltään että muokattavuudeltaan. Järjestelmä tulee helpottamaan jokapäiväistä työskentelyä sen yksinkertaisen ja nopean käyttöliittymän ansiosta.

## ABSTRACT

Author	Nico Kauppila
Title	A CRM-system
Year	2018
Language	Finnish
Pages	28
Name of Supervisor	Sirkka Hellman

---

This thesis focused on Microsoft O365 CRM solution built on the system. The programming used the AngularJS library as well as the Bootstrap HTML base. The work was commissioned by Pilvilampi Software Oy.

The goal of the bachelor's degree thesis was to create a system for maintaining sales, contracts, projects and Helpdesk service. With this system important business information will be in one place and work becomes easier because all the information is quickly accessible to everyone.

The case company has previously used ProjectWeb for the same purpose, but because of its complexity and limited customizability, the aim to create a complete system that could be updated and adapted to the company's changing needs. The result of the work was a system that was better than its predecessor, both in terms of usability and formability. The system will make every day work much easier with its simple and fast user interface.

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	7
1.1	Toimeksiantaja.....	7
1.2	Tavoitteet .....	7
2	OFFICE 365 / SHAREPOINT .....	9
2.1	Kentät.....	9
2.2	Sisältötyypit .....	10
2.3	Listat .....	10
2.3.1	Metatiedot .....	11
2.4	REST-hakurajapinta.....	12
2.4.1	Getbytitle.....	13
3	AMAZON AWS.....	14
3.1	AWS API-Gateway.....	14
3.2	AWS S3 .....	15
3.3	AWS CloudFront .....	15
4	PILVILAMPI SOVELLUSALUSTA .....	16
4.1	AngularJS.....	16
4.1.1	Angular-formly .....	17
4.1.2	Angular-Datatables .....	18
4.2	Node.js .....	19
4.3	Microsoft Flow.....	19
5	JÄRJESTELMÄN TOTEUTUS .....	21
5.1	Yleiskuvaus.....	21
5.2	Sivuston rakenne ja käyttöliittymä.....	23
6	YHTEENVETO .....	27
	LÄHTEET.....	28

**KÄSITTEET**

<b>Office 365</b>	Microsoftin pilvipalvelu.
<b>AWS</b>	Amazon Web Services pilvipalvelu
<b>AWS API-Gateway</b>	Palvelu, jonka avulla kehittäjät voivat luoda, julkaista, ylläpitää, seurata ja suojata sovellusliittymiä
<b>AWS S3</b>	Amazonin tallennuspalvelu
<b>AWS CloudFront</b>	Verkkopalvelu, joka nopeuttaa verkkosisällön jakamista.
<b>AngularJS</b>	Avoimen lähdekoodin JavaScript-ohjelmistokehys
<b>NodeJS</b>	Avoimen lähdekoodin JavaScript moottoriin perustuva serveripuolen sovellusalusta.
<b>CRM</b>	Asiakkuudenhallinta-järjestelmä (Customer Relationship Management)
<b>BootStrap</b>	Twitterin kehittämä HTML5 käyttöliittymäpohja
<b>IaaS</b>	Infrastructure as a Service eli infrastruktuuria palveluna, jossa asiakkaalle tarjotaan web-pohjainen hallintaliittymä käyttöön.
<b>PaaS</b>	Sovellusalusta palveluna.
<b>SaaS</b>	Ohjelmisto palveluna kokonaisuudessaan.

## KUVAT

Kuva 1. Esimerkki Sharepointin kentästä, joka on määritelty yksiriviseksi.....	9
Kuva 2. Esimerkki sisältötyypistä, joka sisältää kenttiä. ....	10
Kuva 3. Esimerkki SharePointin listanäkymästä. ....	11
Kuva 4. Sharepoint-listan näkymä metatiedoista.....	11
Kuva 5. Sharepoint-listan metatiedot.....	12
Kuva 6. API-Gatewayn toimintaperiaate .....	14
Kuva 7. Esimerkki AngularJS:n käytöstä Invoice-lomakkeella. ....	17
Kuva 8. Esimerkki Angular-formly:n lomakekentästä, joka luo lomakkeelle Title- kentän. ....	18
Kuva 9. Datatables-esimerkki Invoice-lomakkeen laskunäkymästä.....	18
Kuva 10. Esimerkki node.js:n käytöstä.....	19
Kuva 11. Task -listalle tehty flow sähköpostin lähettämisestä. ....	20
Kuva 12. Kuva sovelluksen dashboard-näkymästä.....	23
Kuva 13. Kuva sovelluksen navigaatiopalkista .....	24
Kuva 14. Task-listan filteröidyt tabit .....	24
Kuva 15. Mobiilinäkymä. ....	25
Kuva 16. Näkymä sivustolla olevasta Contacts-listasta.....	26

# 1 JOHDANTO

Opinnäytteessä kuvataan työorganisaation kehittämistoimenpide uuden asiakkuudenhallintajärjestelmän muodossa. Projektin tarkoituksena on luoda ja käyttöönottaa uusi järjestelmä, joka parantaa yrityksen sisäistä ja ulkoista toimintaa, laskutuslogiikkaa, sidosryhmäviestinnän laatua, nopeutta ja luotettavuutta. Samalla järjestelmä kehittää ja suoraviivaistaa palvelutoimitusketjua sekä nopeuttaa asiakasprojektien läpimenoaikoja.

Kehitettävä järjestelmä tulee muuttamaan merkittävästi työpaikan toimintatapoja ja tehostamaan työ-, organisaatio- ja johtamiskäytäntöjä. Keskeiset tavoitteet järjestelmän implementoinnissa ovat parantaa projektien hallintaa, työtehtävien kirjausta sekä työtuntien kirjausta. Tämän lisäksi konkreettiset tavoitteet ovat dokumentinhallinnan kehittäminen sekä laadun seuranta järjestelmästä saatavan tiedon perusteella.

## 1.1 Toimeksiantaja

Työn toimeksiantajana toimii vaasalainen yritys Pilvilampi Software Oy. Yritys on Amazon AWS, Sharepoint ja O365-ratkaisuihin keskittynyt ohjelmistoyritys. Yrityksen toimialue ulottuu käytännössä ympäri maailmaa eli asiakaskunta on hyvin laaja.

## 1.2 Tavoitteet

Työn tavoitteena on tehdä oma, toimiva CRM eli asiakkuudenhallintajärjestelmä, joka tulisi yrityksen sisäiseen käyttöön. Se parantaisi ja helpottaisi koko yrityksen työskentelyä huomattavasti, sillä vastaisuudessa kaikki tulevat projektit, dokumentit, asiakkuudet sekä laskutukset löytyisivät yhdestä paikasta. Yrityksellä on aiemmin ollut käytössä Projectweb, joka oli sinänsä toimiva järjestelmä mutta hieman puutteellinen ja sekava. Tuleva järjestelmä tulee tekemään työskentelystä siis huomattavasti helpompaa ja organisoidumpaa.

Menestyksestä hanketta varten on tehty esimäärittely toteutettavasta järjestelmästä ja toteutettu toimiva prototyyppi valitulla teknologialla. Projektin esivaiheessa toteutettiin järjestelmä-, liiketoiminta- ja tuotannollis-tekni- ninen rajapintojen määrittely, tuotannon katselmoinnit sekä varsinaisen prototyypin ohjelmointi.

Nyt suunnitellussa hankkeessa toteutetaan prototyypin kehittäminen toimivaksi järjestelmäksi sekä suoritetaan läpi organisaation toteutuva ohjelmiston käyttöönotto. Vuoden 2017 aikana suoritettussa esiselvityksessä on kartoitettu käytettävien tietoteknisten järjestelmien vaatimusmäärittely ja arkkitehtuuri, määriteltävä vaadittavat integrointirajapinnat sekä laadittu niiden perusteella toteutettavan järjestelmän toteutussuunnitelma.

Projektin konkreettisenä tavoitteena on toteuttaa ja implementoida tietotekninen asiakkuudenhallintajärjestelmän ominaisuuksineen, joka standardoi projektien johtamisen ja niiden vaatiman viestinnän ja dokumentoinnin, käytettävän tuotantojärjestelmän sekä palvelulaskutuksen.

Kehitysprojekti pitää sisällään valmisteluhankkeessa toteutetun proto-version implementoinnin O365- järjestelmään. Hankkeen tavoitteena on luoda järjestelmältä vaadittavat ominaisuudet, ja käyttöohjeistus sekä esitetyn rautalankamallin mukainen järjestelmä.



## 2 OFFICE 365 / SHAREPOINT

Office 365 on palvelu, jonka ympärille kaikki rakennetaan - ei pidä siis sekoittaa Officea ja SharePointia keskenään, sillä ne ovat kaksi eri ohjelmistoa. Office 365:seen on mahdollista hankkia maksullisella lisenssillä yllämainitun SharePointin ja lukuisia muita työkaluja, jotka helpottavat yrityksen ja myös kotikäyttäjänkin päivittäistä työtä.

SharePoint mahdollistaa oman tietoturvan, työryhmien, asiakirjojen ja tietojen hallinnan osana Office365-pakettia. Tässä tulemme näyttämään, kuinka SharePointia voi käyttää perinteisen Intranet -sivuston sijaan muussakin tarkoituksessa ja kuinka sen päälle on mahdollista konfiguroida oma sovellus. (Alexander, 2017)

### 2.1 Kentät

Kentät ovat SharePointin listojen metatietokenttiä. Kentät voivat olla sidoksissa sisältötyypeille tai suoraan listoihin. Tiedontalennuksessa on oltava tarkkana kenttien nimeämisen kanssa, sillä ääkköset ja erikoismerkit sotkevat sen ns. ”internal” nimen, joka saattaa aiheuttaa koodipuolella vaikeuksia (Kuva 1). Myöskin liian pitkiä nimiä olisi viisasta välttää samasta syystä. (O’Connor 2016, 4-5)

Name and Type  
Type a name for this column.

Column name:  
FirstName

The type of information in this column is:

- Single line of text
- Multiple lines of text
- Choice (menu to choose from)
- Number (1, 1.0, 100)
- Currency (\$, ¥, €)
- Date and Time

Group  
Specify a site column group.  
Categorizing columns into groups will make it easier for users to find them.

Put this site column into:

- Existing group:  
CRM
- New group:

**Kuva 1.** Esimerkki Sharepointin kentästä, joka on määritelty yksiriviseksi.

## 2.2 Sisältötyypit

Sisältötyypillä (Content Type) on erittäin tärkeä rooli Sharepointissa. Se määrittelee kuinka Sharepointin sivut, dokumentit ja muut listan objektit luokitellaan. Sisältötyyppiä voidaan käyttää Sharepointin REST-hakutoiminnoissa rajaamaan haettavaa tietoa sisältötyypin perusteella. Sisältötyypit ovat joukko sarakkeita, jotka on ryhmitelty yhteen tiettyyn tarkoitukseen (Kuva 2). Voit esimerkiksi luoda sisältötyypin (ryhmä sarakkeita eli kenttiä), joka nimetään "Laskut", jonka jälkeen voit lisätä sen jokaiseen listaan tai kirjastoon, joka tarvitsee tällaisia tietoja. (Microsoft Dev Center 2017a)

Columns	Name	Type	Status
	Title	Single line of text	Required
	AccountNumber	Single line of text	Optional
	BusinessID	Single line of text	Required
	Address	Multiple lines of text	Optional
	City	Single line of text	Optional
	PostalCode	Single line of text	Optional
	wwwAddress	Single line of text	Optional
	PrimaryContact	Lookup	Optional
	Comment	Multiple lines of text	Optional
	Owner0	Person or Group	Optional
	HourlyRate	Currency	Optional
	Customer	Choice	Optional
	PaymentTerms	Single line of text	Optional

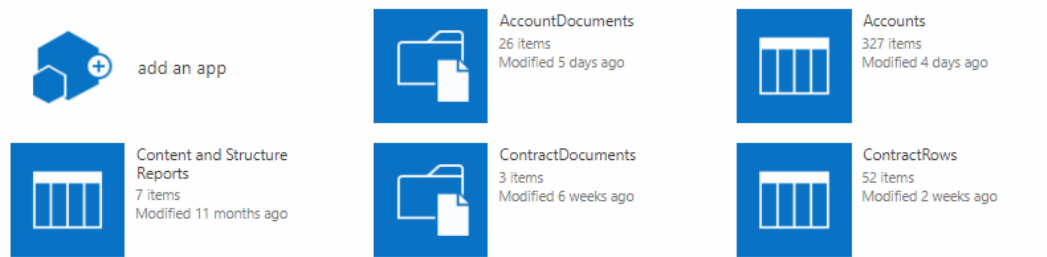
**Kuva 2.** Esimerkki sisältötyypistä, joka sisältää kenttiä.

## 2.3 Listat

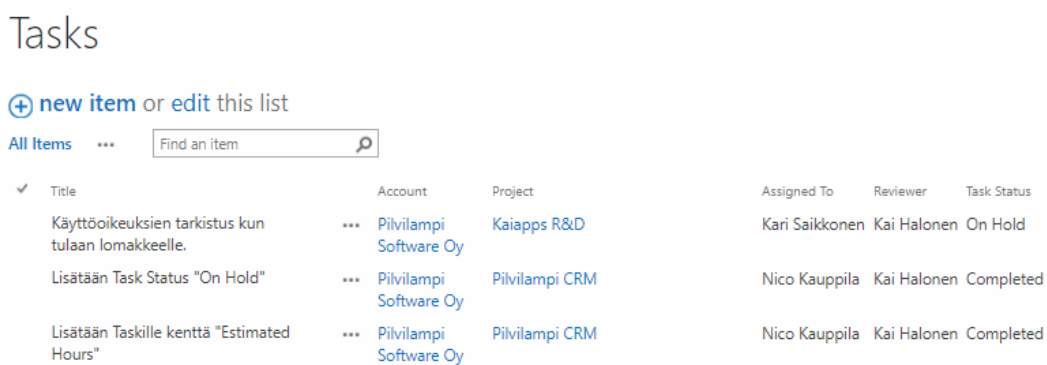
Listat eli dokumenttikirjastot ovat Sharepointin käyttämiä listoja, joiden avulla tiedot voidaan näyttää sarakkeittain ja riveittäin (Kuva 3). Tämän lisäksi sivuston tekijällä on myös mahdollista määritellä omia mukauttamiaan listoja ja luoda niille näkymät.

Listat voivat sisältää monia erilaisia tietoja, jotka vaihtelevat tekstistä ja päivämäärästä, muuhun sarakkeeseen perustuviin laskelmiin (Calculated field). Voit asettaa hälytyksiä, saada ilmoituksia tai tilata RSS-syötteen (Really Simple Syndication), joka sisältää otsikon ja URL-osoitteen muuttuville listan tiedoille (Kuva 4). (Support.office.com 2017)

## Lists, Libraries, and other Apps



**Kuva 3.** Esimerkki SharePointin listanäkymästä.





**Kuva 4.** Sharepoint-listan näkymä metatiedoista.

### 2.3.1 Metatiedot

Metatiedot esittävät dokumenttien eli listojen tietoja. Näillä tiedoilla on vaikutus dokumenttien hallinnassa ja informaation rakenteessa (Kuva 5). Näitä metatietoja ovat esimerkiksi

- ryhmittely
- säännöt ja toimintalogiikka
- haku
- käyttöoikeudet

Metatietotyypeillä on lisäksi omat metatietotyyppinsä - yleisimpiä niistä ovat totta-kai otsikko, päivämäärä, käyttäjä ja ID. (Wade, 2017)

Title *	<input type="text"/>
Account	<input type="text" value="(None)"/>
Project	<input type="text" value="(None)"/>
Assigned To	<input type="text" value="Enter a name or email address..."/>
Reviewer	<input type="text" value="Enter a name or email address..."/>
Task Status	<input type="text"/>
TaskType	<input type="text"/>
Due Date	<input type="text"/> 
Priority	<input type="text" value="(2) Normal"/>
Start Date	<input type="text" value="22.4.2018"/> 
EstimatedHours	<input type="text"/>
Comment	<div style="border: 1px solid #ccc; height: 60px;"></div>

**Kuva 5.** Sharepoint-listan metatiedot

## 2.4 REST-hakurajapinta

Sovelluksessa on käytetty hyödyksi REST (REpresentational State Transfer) -rajapintaa. REST käyttää HTTP-protokollan eri toimintoja eli metodeja. HTTP-protokollan metodit ovat GET, POST, PUT ja DELETE. Tehdyssä sovelluksessa on käytetty pelkästään GET- ja POST-metodeja. GET-metodi tarkoittaa tiedon hakemista palvelimelta ja POST-metodi tiedon lähettämistä palvelimelle. (Microsoft Dev Center 2017b)

Sharepointin REST-tuki tekee mahdolliseksi Sharepoint-objektien käyttämisen REST-kutsujen avulla. REST-kutsut tehdään käyttämällä osoitetietoa ja määrittelemällä REST-kysely selaimen osoiteriville. Hakuvaihtoehtoja ovat esimerkiksi listakohtainen haku tai mahdollisesti koko sivuston haku. (Microsoft Dev Center 2017b)

[https://demo.sharepoint.com/sites/DemoCRM/\\_api/web/](https://demo.sharepoint.com/sites/DemoCRM/_api/web/)

### 2.4.1 Getbytitle

Sharepointissa olevat REST-kyselyt voidaan ohjata hakemaan suoraan listalta käyttämällä 'getbytitle'-komentoa. Tämän komennon avulla pystytään hakemaan listalla olevia sarakkeita käyttämällä listan (dokumenttikirjaston) nimeä hakusanana rajaamaan tuloksia. (Microsoft Dev Center 2017c)

[https://demo.sharepoint.com/sites/DemoCRM/\\_api/web/Lists/getbytitle\('Accounts'\)/Items?\\$select=Id,Title](https://demo.sharepoint.com/sites/DemoCRM/_api/web/Lists/getbytitle('Accounts')/Items?$select=Id,Title)

Kuvan komento hakee kaikki kyseisellä "Accounts" listalla olevat "items" eli ne sarakkeet, joissa on kyseisen listan tiedot sekä näyttää sieltä kaikki "id" ja "title" -kentät. (Microsoft Dev Center 2017c)

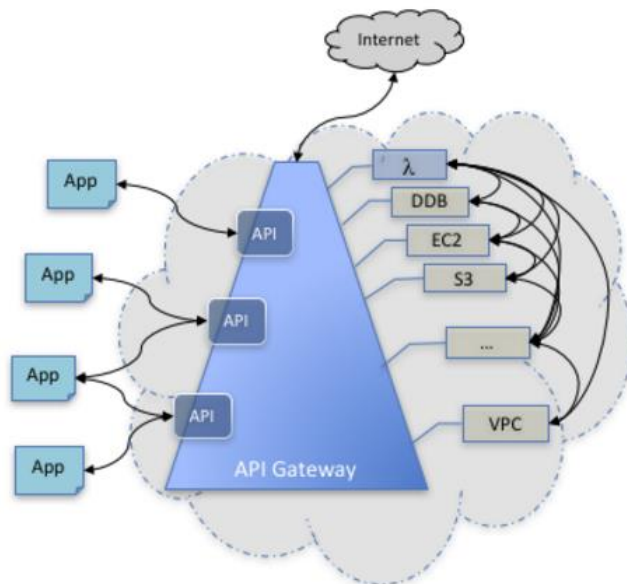
### 3 AMAZON AWS

Amazon Web Services on iso pilvipalvelu täynnä erilaisia etätietojenkäsittely-resurssien palveluja. AWS-pilvipalvelu soveltuu niin IaaS, PaaS kuin SaaS-käyttöön. Pilvipalveluiden käyttöönottamiseen ja hallintaan on oma hallintapaneelinsa, josta löytyy tarvittavat työvälineet konfiguroimiseen sekä sovelluskehitykseen.

#### 3.1 AWS API-Gateway

Ohjelmistorajapinnan välityspalvelin, jonka avulla kehittäjillä on mahdollisuus luoda, julkaista, ylläpitää, seurata ja suojata sovellusliittymiä. API-Gateway toimii kommunikaatiossa yhdessä sovelluksen kanssa tarjoten sille sovellusrajapinnan, joka mahdollistaa datan tallentamisen ja liikkumisen molempiin suuntiin (Kuva 6). (Amazon AWS, What is amazon API Gateway?)

Tekemäni sovelluksen konfiguraatitiedostojen hakeminen tapahtuu näin myöskin API-Gatewayn:n kautta, joka mahdollistaa datan tallentamisen ja hakemisen vauhtomasti ja nopeasti ilman suuria aikaviiveitä, joka tällaisessa sovelluksessa on hyvin tärkeää käytettävyyden kannalta.



**Kuva 6.** API-Gatewayn toimintaperiaate

### 3.2 AWS S3

AWS S3 on tietojen tallentamiseen ja jakamiseen rajapinnan yli tarkoitettu skaalautuva tallennustilapalvelu, joka sisältää niin sanottuja ”bucketeja” eli säiliöitä. Se on helppo integroida erilaisiin palveluihin avointen REST- ja SOAP-rajapintojen avulla ja sitä voi myös käyttää normaalin kiintolevyn tavoin. Sen voi myöskin liittää yhtä aikaa useampaan palvelimeen, mikäli tilaa tarvitsee enemmän tai dataa täytyy siirrellä paljon paikasta toiseen. (Amazon Simple Storage Service Developer Guide API Version 2006-03-01)

### 3.3 AWS CloudFront

Amazon CloudFront on verkkopalvelu, joka nopeuttaa staattisen ja dynaamisen verkkosisällön, kuten .html-, .css-, .js -tiedostojen jakelua käyttäjille. CloudFront toimittaa sisällön isojen datakeskusten verkkojen kautta, joita kutsutaan niin sanotuksi ”reunakohteiksi”. Kun käyttäjä pyytää CloudFront-palvelua tarjoavaa sisältöä, käyttäjä ohjataan reitin sijaintiin, joka sitten tarjoaa pienimmän latenssin (aika-viive), jonka kautta sisältö voidaan toimittaa parhaalla mahdollisella suorituskyvyllä.

CloudFront nopeuttaa sisällön jakamista reitittämällä kunkin käyttäjän pyynnön reunapaikkaan, joka parhaiten sopii sisällölle, eli sijaitsee lähimpänä tai tarjoaa nopeimman toimituksen käyttäjälle. Tämä vähentää huomattavasti verkkojen määrää, joita käyttäjien pyyntöjen on läpäistävä, mikä takaa suuremmat tiedonsiirtonopeudet datalle.

CloudFront -tekniikka parantaa huomattavasti sisällön luotettavuutta ja saatavuutta, sillä kopiot tiedostoista säilytetään välimuistissa useilla reuna-alueilla ympäri maailmaa. (Amazon CloudFront Developer Guide API Version 2016-09-29)

## 4 PILVILAMPI SOVELLUSALUSTA

Pilvilampi sovellusalusta on konfiguroitava järjestelmä, jolla voidaan helposti ja nopeasti rakentaa yrityksille erilaisia O365 sovelluksia. Järjestelmän tavoite on vähentää yritysten tarvetta erillisille pilvipalvelu ratkaisuille, jotka eivät välttämättä osaa keskustella keskenään ja yrityksen tärkeät tiedot levittäytyvät eri palveluihin.

Pilvilammen ratkaisussa haluttu järjestelmä voidaan rakentaa asiakkaan omaan ympäristöön ja tärkeät tiedot ovat aina asiakkaan hallussa. Järjestelmän käyttöönotto vähentää VarjoIT:tä ja näin ollen helpottaa yrityksen CIO:n (Chief Information Officer) työtä. Pilvilampi on toteuttanut järjestelmiä myynnin- ja projektinhallintaan, sekä kiinteistöjenhallintaan ja kunnossapitoon.

### 4.1 AngularJS

AngularJS on vuonna 2012 julkaistu Googlen ylläpitämä avoimen lähdekoodin JavaScript-ohjelmistokehys, joka on tarkoitettu ensisijaisesti yksisivuisten web-sovellusten kehittämiseen. AngularJS:n tavoite on lisätä selainpohjaisiin sovelluksiin tuki MVC-arkkitehtuurille (Model-View-Controller) eli malli-näkymä-kontrolleri, jonka avulla sivustojen kehitys ja testaus helpottuu huomattavasti. Se on täysin laajennettavissa ja toimii hyvin muiden kirjastojen kanssa. Sen ominaisuudet ovat myöskin helposti muokattavissa projektiin ja siihen vaadittuihin ominaisuuksiin sopiviksi (Kuva 7). AngularJS on saanut nimensä aaltosulkeista (`{ }`).

AngularJS -kirjaston toiminnallisuus perustuu siihen, että se lukee HTML-elementteihin tehtyjen attribuuttien kautta, minkälaisia toimintoja halutaan tehdä, jonka jälkeen se yhdistää HTML-elementtiin tehdyn toiminnon liitettynä siinä oleviin muuttujiin. Näitä muuttujia on mahdollista käyttää staattisesti eli asettamalla haluttu arvo muuttujaan JavaScriptiä käyttäen tai vuorostaan dynaamisesti JSONilla. (AngularJS.org 2017)



```

<div class="row">
  <div class="col-xs-12">
    <table class="table table-striped table-hover">
      <thead>
        <tr>
          <th> # </th>
          <th> Item </th>
          <th class="hidden-xs"> Description </th>
          <th class="hidden-xs"> Quantity </th>
          <th class="hidden-xs"> Unit Price </th>
          <th> Total</th>
        </tr>
      </thead>
      <tbody ng-init="total = 0">
        <tr ng-repeat="item in genericDataCtrl.data">
          <td>{{index+1}} </td>
          <td>{{item.Title}} </td>
          <td>{{item.Description}} </td>
          <td>{{item.Qty}} </td>
          <td>{{item.Price | number: 2}} EUR </td>
          <td ng-init="$parent.total = $parent.total + (item.Qty * item.Price)">{{item.Qty * item.Price | number: 2}} EUR</td>
        </tr>
      </tbody>
    </table>
  </div>
</div>

```

**Kuva 7.** Esimerkki AngularJS:n käytöstä Invoice-lomakkeella.

#### 4.1.1 Angular-formly

Angular-formly on AngularJS:än oma moduuli, jonka avulla voidaan tehdä mukautettuja lomaketoimintoja (Kuva 8). Sen avulla voidaan määritellä lomakekomponentit useissa kohteissa, eikä eksplisiittisesti HTML-dokumentissa. Siitä on monia etuja, kuten helpommin ylläpidettävää ja selkeämpää koodia, mutta myöskin helpottaa validointia juuri sen helpon muokattavuuden ansiosta. (Ryan, 2017)

Angular-formly tukee myöskin JavaScriptiä, joka mahdollistaa monimutkaistenkin lomaketoimintojen tekemisen. Käyttäjän on mahdollista esimerkiksi yhdistää useita kenttiä toimimaan keskenään.

```

{
  "className": "col-md-6",
  "key": "Title",
  "type": "horizontal-Input",
  "templateOptions": {
    "labelClass": "col-md-3 control-label",
    "divClass": "col-md-9",
    "label": "Account",
    "placeholder": "Account",
    "required": true,
    "addonLeft": {
      "class": "glyphicon glyphicon-briefcase"
    }
  }
},
},

```

**Kuva 8.** Esimerkki Angular-formly:n lomakekentästä, joka luo lomakkeelle Title-kentän.

#### 4.1.2 Angular-Datatables

Datatables on AngularJS-laajennus, joka mahdollistaa jQuery Datatables-liitännäisen käytön AngularJS-ympäristössä. Se näyttää datan siististi tauluissa mahdollistaen erilaisten ominaisuuksien, esimerkiksi listan tietojen kokoamisen tässä tapauksessa laskuksi (Kuva 9). Datatables siis laajentaa ja tuo paljon lisäominaisuuksia HTML-taulukon elementteihin.

Datatablesin erilaisilla toiminnoilla voidaan määritellä esimerkiksi, miten taulu järjestää tiedot tai kuinka monta saraketta näytetään näkymässä taikka filteröidään päivämäärän mukaan tulevat tapahtumat, mikäli listalta löytyy päivämäärä-kenttä.

The screenshot shows a web application interface for an invoice. At the top, there is a navigation bar with 'INVOICE', 'INVOICE ROWS', 'PRINT', and 'DOCUMENTS'. Below this, the title 'Demo Asiakas - demoprojekti' is displayed. The main content area is divided into two columns: 'Client:' and 'Info:'. The 'Client:' column contains details such as 'Account number: Demo', 'Account: Asiakas', 'Contact: Matti Meikalainen', 'Email: matti.meikalainen@demo.com', and 'Payment terms: 14'. The 'Info:' column contains 'Esimerkkilasku'. Below this, there is a table with the following data:

#	Item	Description	Quantity	Unit Price	Total
1	Esimerkkilasku	Esimerkkilasku	10	100,00 EUR	1,000,00 EUR

At the bottom left, there is a contact information box for 'Pilvilampi Software Oy' with address 'Vaasanpuistikko 17 9krs, 65100 Vaasa' and contact 'Kai Halonen, kai.halonen@pilvilampi.fi'. At the bottom right, there is a summary section:

**Subtotal:** 1,000.00 EUR  
**VAT:** 24 %  
**Total with VAT:** 1,240.00 EUR

**Kuva 9.** Datatables-esimerkki Invoice-lomakkeen laskunäkymästä.

## 4.2 Node.js

Node.js on avoimen lähdekoodin JavaScript-ympäristön tueksi tehty serveripuolen sovellusalusta, joka antaa rajapinnan ja mahdollistaa JavaScriptin ajamisen palvelimella (Kuva 10). Se käyttää Googlen V8 JavaScript -virtuaalikonetta, joka mahdollistaa raskaiden paljon dataa kuin kevyeen nopeaan käyttöönkin tarkoitettujen projektien läpiviennin. Node.JS:ää käyttävät myös monet suuret yritykset kuten Microsoft ja Mozilla omissa projekteissaan.

Node.JS helpottaa huomattavasti ns. full-stack kehitystä, sillä kehittäjä voi tehdä sekä front-end:iin -että back-end:iin haluttuja ominaisuuksia. (NodeJS.org 2017)

```
3513 var AWS = require('aws-sdk');
3514 var s3 = new AWS.S3();
3515 var myBucket = 'TestBucket';
3516 var myKey = 'TestKey';
3517
3518 s3.createBucket({Bucket: myBucket}, function(err, data) {
3519
3520   if (err) {
3521
3522     console.log(err);
3523
3524   } else {
3525
3526     params = {Bucket: myBucket, Key: myKey, Body: 'Heihei!'};
3527
3528     s3.putObject(params, function(err, data) {
3529
3530       if (err) {
3531
3532         console.log(err)
3533
3534       } else {
3535
3536         console.log("Tiedot siirretty Bucketiin");
3537       }
3538     });
3539   }
3540 });
```

**Kuva 10.** Esimerkki node.js:n käytöstä.

## 4.3 Microsoft Flow

Microsoft Flow:n avulla voidaan automatisoida sivuston toimintoja helpottamaan työnkulkua, sekä päivittämään tietoja automaattisesti tietyin väliajoin niin halutesaan. Sitä voidaan esimerkiksi käyttää siihen, että luetaan sähköpostista tietyt kentät ja päivitetään ne automaattisesti SharePointin listalle (Kuva 11).

Flow name When a new email arrives -> Task ✓ Update flow ✕ Close

The image shows a Microsoft Flow configuration window. At the top, the flow name is "When a new email arrives -> Task". There are two buttons: "Update flow" (checked) and "Close" (marked with an X). The main area contains two steps:

- Step 1: When a new email arrives**
  - Folder: Task
  - Show advanced options: (dropdown arrow)
- Step 2: Create item**
  - \* Site Address: [Redacted URL]
  - \* List Name: Tasks
  - \* Title: Subject
  - Account Id: [dropdown]
  - Project Id: [dropdown]
  - Assigned To Claims: CC
  - Reviewer Claims: From
  - Task Status Value: Todo
  - TaskType Value: New
  - Priority Value: (2) Normal
  - Start Date: Received Time
  - Due Date: [empty]
  - EstimatedHours: [empty]
  - Comment: Body
  - FullName: [empty]
  - Content type Id: [dropdown]

At the bottom, there are two buttons: "+ New step" and "Save flow".

**Kuva 11.** Task -listalle tehty flow sähköpostin lähettämisestä.

## 5 JÄRJESTELMÄN TOTEUTUS

### 5.1 Yleiskuvaus

Järjestelmällä on useita eri käyttäjäryhmiä, joiden tarpeet poikkeavat toisistaan. Yrityksessä on tunnistettu järjestelmälle keskeiset sisäiset ja ulkoiset asiakkaat: Järjestelmäsuunnittelijoilta, projektipäälliköiltä ja HelpDesk-työntekijöiltä edellytetään tehokasta tuntikirjanpitoa eri tehtäviin käytetyn ajan merkitsemiseksi tiettyyn tehtävään sekä asiakas- ja projektikohtaista tuntikirjanpitoa.

Tuotantohenkilöstön lisäksi merkittävässä roolissa on järjestelmän merkitys yritysjohdon johtamistyökaluna. Uusi järjestelmä kehittää myös yrityksen taloushallintoa ja mahdollistaa aikaisempaa tarkemman johdon raportoinnin ja projektien kannattavuuslaskelmat sekä kehittämää laskutusta. Lisäksi asiakkaat ovat keskeisessä asemassa uutta järjestelmää suunniteltaessa. Asiakkaiden osallistaminen projektien hallintaan ja niiden toteutumiseen saadaan järjestelmän avulla myös toteutettua aikaisempaa mutkattomammin. Asiakaspalautteen antaminen järjestelmän kautta kehittää reaaliaikaisesti myös yrityksen laatua.

Tärkeimmät ominaisuudet uudelle järjestelmälle tulevat olemaan uusien työtehtävien eli ns. tehtävien luomisen helpottaminen sekä tehtävölogiikan kehittäminen. Näiden lisäksi järjestelmään luodaan muita uusia projektihallintaa parantavia toiminnallisuuksia, kuten henkilö resurssointi. Järjestelmäsuunnittelijat käyttävät järjestelmää jatkuvasti työtehtävissään, tämä tarkoittaa useita kertoja päivässä. Tärkein ominaisuus tälle käyttäjäryhmälle on nähdä helposti, mihin työtehtävään heidän on tartuttava seuraavaksi.

Työtehtävien nopeaan suorittamiseen he tarvitsevat pikaisen yleiskatsauksen yksittäisistä tehtävistä, jotka ovat heille merkittäviä. Työssään järjestelmäsuunnittelijat tulevat käyttämään suuren osan ajastaan yksittäisten tehtävien, yhteenvetojen sekä tehtäväkuvaussivujen tarkasteluun. Suunnittelijoilla on myös usein tarve lisätä uusia dokumentteja järjestelmään. Tällä käyttäjäryhmällä tuntikirjauksen merkitys on erittäin suuri. Projektipäälliköillä on tavallisesti tietty määrä asiakkaita ja

asiakkuuksiin liittyviä käynnissä olevia projekteja. Tämä käyttäjäryhmä on myös pääsääntöisesti perillä eri projektien tiloista.

Tärkein järjestelmän ominaisuus projektipäälliköille on mahdollisuus luoda uusia tehtäviä ja seurata niiden statusta. Kuten järjestelmäsuunnittelijat, projektipäälliköt käyttävät suuren osan työajasta yhteenvetojen, tehtävälistojen parissa sekä yksittäisten tehtävien tarkastelussa. Suunnittelijoista poiketen, projektipäälliköiden vastuulla on uusien tehtävien luonti järjestelmään. Projektipäälliköiden tulee myös saada lisättyä järjestelmiin erilaisia dokumentteja.

Projektityössä yllämainitun ryhmän kiinnostuksena on myös saada tietoa erilaisten muutosten tilasta sekä erilaisiin tehtäviin ja muutoksiin tarvittavan työn laadusta ja määrästä. Tärkeä ominaisuus on tietää, onko jokin tietty tehtävä laskutettavaa työtä, josta asiakas maksaa tehdyistä työtunneista vai onko kyseessä virheiden korjaukseen liittyvä tehtävä, josta ei voida laskuttaa. Projektipäälliköiden tehtävänä on hyväksyä tehdyt työtunnit, jotka siirtyvät tämän jälkeen laskutukseen.

HelpDesk toimii pääosin samalla tavalla kuin projektipäällikkö, mutta HelpDesk ei tarvitse yhtä syvällistä tietoa projektien yksityiskohdista. Projektipäälliköistä poiketen, HelpDeskillä tulee olemaan laajempaa tietoa kaikista projekteista, kuin yksittäisellä projektipäälliköllä. Asiakkaan soittaessa ja kysyessä tietoa, HelpDeskin täytyy nopeasti löytää asiakkaan projekti, siihen liittyvät tehtävät ja pystyä tarkastamaan niiden sen hetkinen tila. Uusien pyyntöjen osalta HelpDeskin täytyy pystyä myös luomaan uusia tehtäviä sekä lisäämään järjestelmään erilaisia dokumentteja.

Tuntikirjauksen merkitys myös HelpDeskissä on keskeisessä roolissa. Tällä hetkellä ulkoistettu virheilmoituslinja vastaanottaa asiakkaiden ilmoittamat virheet. Tämän tehostaminen on myös keskeinen osa uutta järjestelmää. Toissijaisilla käyttäjillä, jotka käyttävät järjestelmää satunnaisesti, on jokin tietty syy kirjautua projektinhallintajärjestelmään. Tällainen tilanne voi olla vaikka, kun asiakas soittaa ja kysyy projektin statusta.

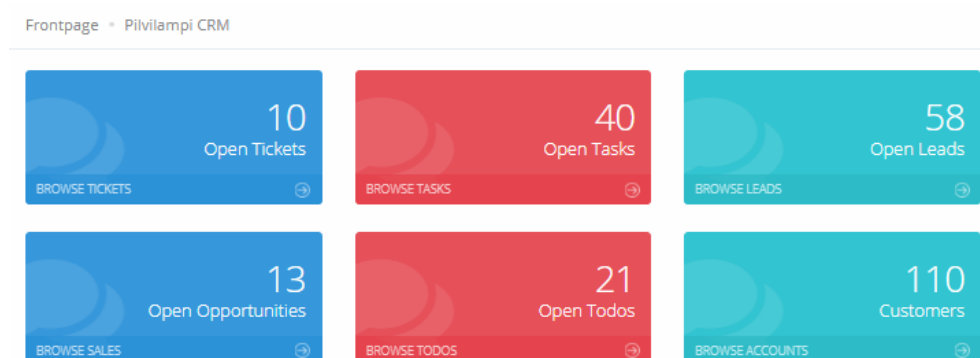
Työntekijällä ei välttämättä ole tietoa kyseisestä projektista ja järjestelmän avulla hän saa nopean kuvan siitä, mistä on kysymys. HelpDeskistä ja projektipäälliköistä poiketen, satunnaiset käyttäjät eivät ole järjestelmän tehokäyttäjiä ja jäävät helposti sen toiminnollisuuksien ulkopuolelle. Tämän käyttäjäryhmän tulee myös päästä kiinni kaikkeen projekti- ja asiakasinformaatioon, kun taas projektipäälliköt ja HelpDesk kaipaavat niitä harvemmin, koska ovat sijoitettuna tiettyihin projekteihin.

Yrityksen ulkoisia asiakasryhmiä ovat asiakkaan projektipäälliköt, hankkeiden ohjausryhmien jäsenet, päivittäiset Pilvilampi Software Oy:n palveluiden käyttäjät sekä alihankkijat. Ulkoisten käyttäjien ensisijaiset käyttötarpeet liittyvät projektien seurantaan, dokumentointiin, budjettiin sekä HelpDesk -tukipyyntöjen tekemiseen.

## 5.2 Sivuston rakenne ja käyttöliittymä

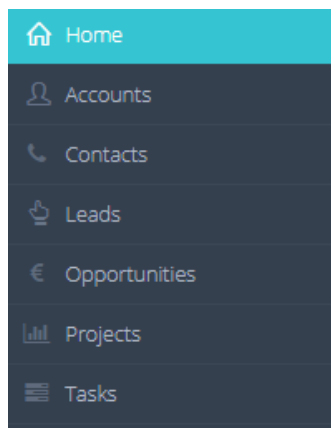
Luotu sivusto koostuu erilaisista AngularJS:än palikoista ja BootStrap-templaa-teista, joita muokkaamalla on saatu tyylikäs mutta simppele sivusto aikaiseksi.

Etusivulta löytyy ”dashboard” eli niin sanottu yleisnäkymä kaikista tärkeimmiksi koetuista asioista. Näkymää voi niin halutessaan muokata dashboard.js -konfiguraatitiedoston kautta (Kuva 12).



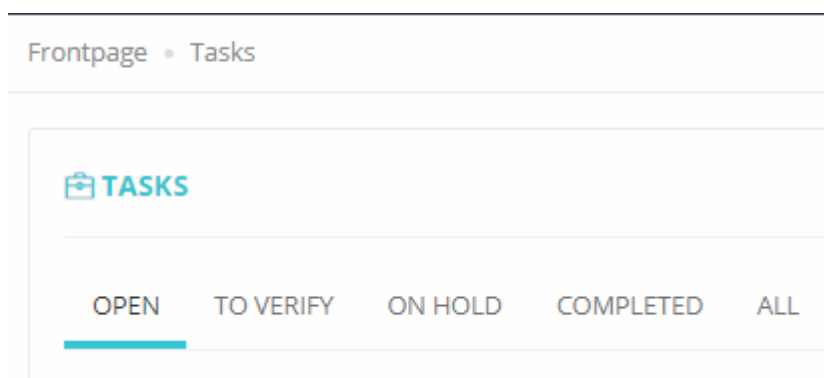
**Kuva 12.** Kuva sovelluksen dashboard-näkymästä

Sivuston vasemmasta laidasta löytyy navigaatio, jonka kautta pääsee käsiksi listoihin ja niiden tietoihin (Kuva 13). Tiedot tallentuvat samalla SharePointin puolelle, mistä ne haetaan ja näytetään.



**Kuva 13.** Kuva sovelluksen navigaatiopalkista

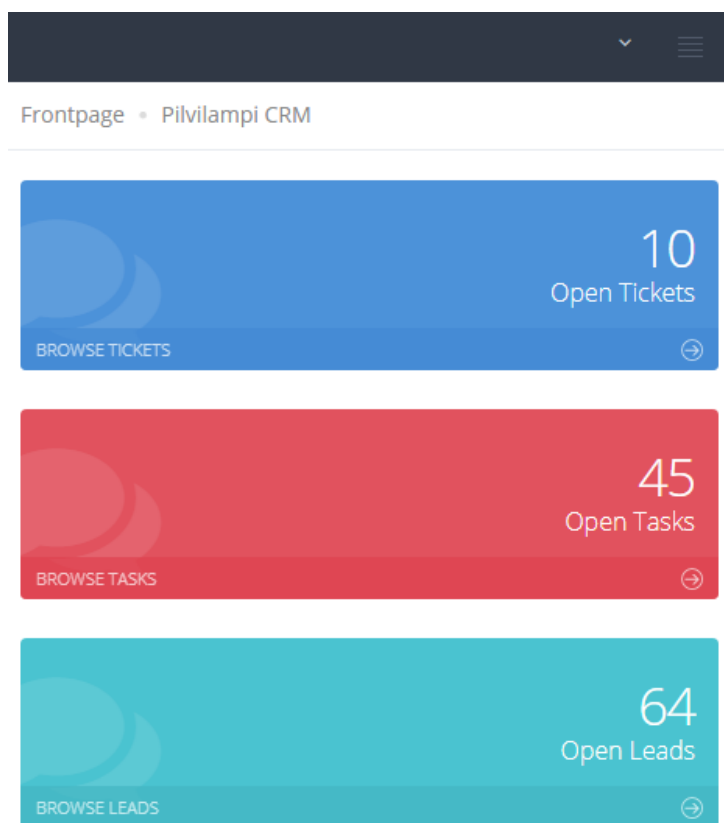
Tiedot ovat myöskin filteröity listoilla, jotta saataisiin helpommin seurattua, missä mennään ja välttyttäisiin kaaokselta (Kuva 14). Listojen filteröinti helpottaa suuresti datan löytämistä ja sujuvoittaa työskentelyä.



**Kuva 14.** Task-listan filteröidyt tabit

Sivuston käyttäminen on helppoa sen yksinkertaisen ulkoasun ja käyttöliittymän takia. Bootstrap -templaatteja käyttäen sivusto on suunniteltu toimimaan niin mobiilissa kuin selaimessakin mahdollistaen responsiivisen käyttökokemuksen. (Kuva 15). Navigaatio -palkki siirtyy mobiiliinäkymän yläpuolelle niin sanotuksi ”burgeriksi”.

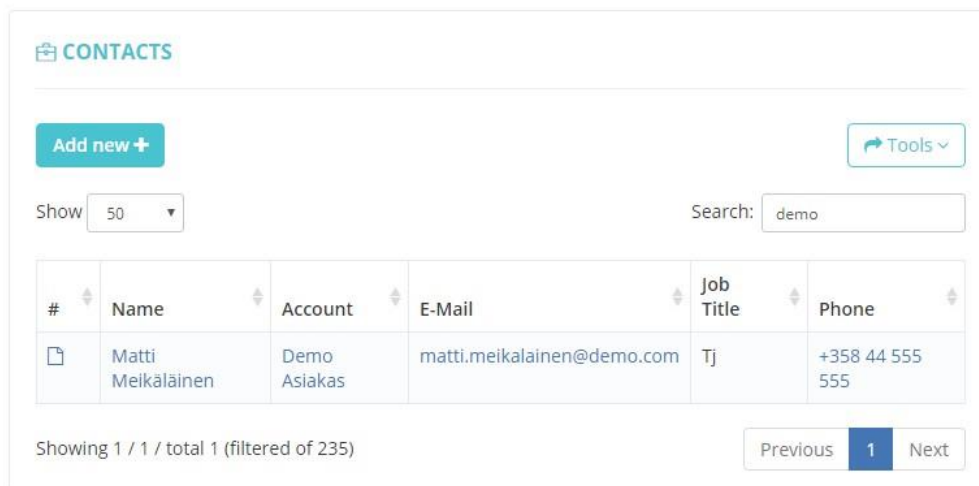




**Kuva 15.** Mobiilinäkymä.

Sovelluksessa on mahdollista määrittellä omia mukautettuja listoja ja luoda niille näkymät (Kuva 16). Listalle voi niin halutessaan tuoda sarakkeita myös muilta listoilta, joka mahdollistaa yhden suhde moneen tai monen suhde moneen periaatteen.

Listat voivat sisältää monia erilaisia tietoja, jotka vaihtelevat tekstistä ja päivämäärästä niin valuutta kuin valinta eli ”choice”-kenttiin.



**CONTACTS**

Add new + Tools ▾

Show 50 Search: demo

#	Name	Account	E-Mail	Job Title	Phone
	Matti Meikalainen	Demo Asiakas	matti.meikalainen@demo.com	Tj	+358 44 555 555

Showing 1 / 1 / total 1 (filtered of 235) Previous **1** Next

**Kuva 16.** Näkymä sivustolla olevasta Contacts-listasta

## 6 YHTEENVETO

Työn tavoitteena oli rakentaa oma, toimiva CRM eli asiakkuudenhallintajärjestelmä, joka tulisi yrityksen sisäiseen käyttöön. Se parantaisi ja helpottaisi koko yrityksen työskentelyä huomattavasti, sillä vastaisuudessa kaikki tulevat projektit, dokumentit, asiakkuudet sekä laskutukset löytyisivät näin yhdestä paikasta. Tuleva järjestelmä tulee tekemään työskentelystä siis huomattavasti helpompaa ja organisoitumpaa.

Tavoite täyttyi ja onnistuimme yrityksessä saamaan uuden toimivan järjestelmän, joka on joka -päiväisessä käytössä ja täyttää tarpeemme kaikilta osin. Jatkossakin tarpeiden muuttuessa –sen helpon muokattavuuden ansiosta. Järjestelmä tai sovellus, miksi sitä haluaakaan kutsua, on tehty AngularJS-ohjelmistokehyksen päälle käyttäen Office365 ympäristöä sekä SharePoint-listoja ja dokumenttikirjastoja.

Projekti piti sisällään paljon uuden oppimista sekä myöskin jo vanhan kehittämistä, sillä en ollut aiemmin tutustunut esimerkiksi AngularJS:ään. Opettelin AngularJS-ohjelmistokehyksen käytön projektia varten, vailla aiempaa kokemusta. Työssä ongelmia aiheutti välillä versioiden muuttuminen AngularJS-versioiden välillä ja se, mitkä toiminnot muuttuivat päivitysten myötä ja voitiin näin tehdä eri tavalla kuin aiemmissa versioissa. Tästä aiheutui välillä paljonkin lisätöitä ja taustatutkimusta työn lomassa.

Projekti itsessään sujui hienosti ilman sen suurempia ponnisteluja tai vaikeuksia. Suurimmat ongelmat sen sijaan tulivat vastaan aikataulutuksessa, sekä opinnäytetyön kirjoituksessa.

## LÄHTEET

Microsoft Dev Center 2017a. Introduction to Content Types. Viitattu 3.9  
[https://msdn.microsoft.com/en-us/library/office/ms472236\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/ms472236(v=office.14).aspx)

Microsoft Dev Center 2017b. REST API Overview. Viitattu 3.9.2017  
<https://msdn.microsoft.com/en-us/library/office/jj163876.aspx>

Microsoft Dev Center 2017c. ListCollection.GetByTitle method. Viitattu 3.9.2017  
<https://msdn.microsoft.com/en-us/library/microsoft.sharepoint.cli-ent.listcollection.getbytitle.aspx>

Matt Wade. Beginners guide to SharePoint metadata. Viitattu 12.9.2017  
<http://icansharepoint.com/beginners-guide-sharepoint-metadata/>

Amazon AWS. Amazon API Gateway. Viitattu 14.9.2017 <http://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

Benjamin Niaulin. What are SharePoint Content Types? Learn How to Create and Use Them. Viitattu 19.9.2017 <https://en.share-gate.com/blog/sharepoint-content-types-understand-use-create>

Microsoft. Create a list in SharePoint. Viitattu 20.9.2017 <https://support.office.com/en-us/article/create-a-list-in-sharepoint-0d397414-d95f-41eb-addd-5e6eff41b083?ui=en-US&rs=en-US&ad=US>

Michael Alexander. What is SharePoint. Viitattu 20.9.2017 <http://www.dummies.com/software/microsoft-office/excel/what-is-sharepoint/>

AngularJS. Developer guide: What is AngularJS? Viitattu 20.9.2017  
<https://docs.angularjs.org/guide/introduction>

NodeJS. Viitattu 20.9.2017 <https://nodejs.org/en/>

Chenkie Ryan. Easy AngularJS Forms with Angular-formly. Viitattu 20.9.2017  
<https://scotch.io/tutorials/easy-angularjs-forms-with-angular-formly>

Amazon Simple Storage Service. Developer Guide (API Version 2006-03-01). Viitattu 18.4.2018 <https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html>

Amazon Simple Storage Service. Developer Guide (API Version 2016-09-29). Viitattu 18.4.2018 <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

Carl O'Connor. SharePoint 101: SharePoint Site Column. Viitattu 25.4.2018  
<https://support.office.com/en-us/article/introduction-to-lists-0a1c3ace-def0-44afb225-cfa8d92c52d7>