Juha Venetjoki

Joiqun jatkokehitys ja integraatiot

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

18.6.2021

Tiivistelmä

Tekijä:	Juha Venetjoki
Otsikko:	Joiqun jatkokehitys ja integraatiot
Sivumäärä:	21 sivua
Aika:	18.6.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Osaamisaluepäällikkö Janne Salonen

Insinöörityön tavoitteena oli tutkia ja jatkokehittää digitaalista työtilaa nimeltä Joiqu. Aluksi tutkittiin työtilan toimivuutta ja toiminallisuuksia. Päätettiin tehdä aluksi Joiqun palvelun siirto toiselle palvelimelle.

Palvelulle oli tarve saada nopeutta ja vakaampi alusta. Palvelu kärsi satunnaisesta hidastelusta. Yhteys tietokantaan oli hidas ja tämä näkyi selkeästi esimerkiksi käyttöliittymän puolella kun paljon tietoa latautui käyttöliittymään.

Kehitystyö keskittyi Joiqun käyttöliittymän uudistamiseen ja integraatioon Severan kanssa. Kehitysversion nimeksi valittiin Joiqu 2.0. Tämän version tarkoituksena oli toimia Visma Severa -järjestelmän viestintäalustana.

Avainsanat: PHP, MySQL, digitaalinen työtila, integraatiot, REST API

Abstract

Author:	Juha Venetjoki
Title:	Further development and integrations of Joiqu
Number of Pages:	21 pages
Date:	18 June 2021
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Software Engineering
Instructors:	Janne Salonen, Head of School (ICT)

The purpose of the thesis was to research and further develop a digital workspace called Joiqu. Initially, the functionalities of the workspace were studied. It was decided to transfer the Joiqu service to another server.

The Joiqu service needed to have more performance and a more stable. The service suffered an occasional poor performance. The connection to the database was slow and this was clearly visible on the user interface side, for example, when a lot of information was loaded into the user interface.

The development work focused on the renewal and integration of Joiqu's user interface with Severa. Joiqu 2.0 was chosen as the name of the development version. The purpose of this version was to serve as a communication platform for the Visma Severa system.

Keywords: PHP, MySQL, digital workspace, integrations, REST API

Sisällys

Lyhenteet

PHP: Lyhenne sanoista Hypertext Preprocessor. Palvelinympäristö, jota käytetään dynaamisten verkkosivujen rakentamisessa.

MySQL: MySQL on relaatiotietokantaohjelmisto.

MVC: Lyhenne sanoista Model-View-Controller. Ohjelmistoarkkitehtuuri, jota käytetään käyttöliittymien suunnittelussa.

CSS: Lyhenne sanoista Cascading Style Sheets. Tyylimäärittelyt verkkosivuelementeille.

JQuery: JQuery on JavaScript-kirjasto, joka on erittäin suosittu syntaksi, mikä on tehty helposti ymmärrettäväksi.

HTML: Verkkosivujen rakentamisessa käytettävä vakioitu merkintäkieli.

REST: Ohjelmointirajapintojen rakentamiseen tarkoitettu arkkitehtuurimalli.

Apache: Palvelinohjelmisto, jota käytetään www-palvelimilla.

Nginx: Palvelinohjelmisto, jota käytetään www-palvelimilla.

Node.js: Palvelimella toimiva JavaScript-koodin suorittamisen ajoympäristö.

React.js: JavaScript-kirjasto, jota käytetään käyttöliittymien rakentamisessa.

Express.js: JavaScript-kehys, jota käytetään esimerkiksi tiedon käsittelyssä rajapintojen välillä.

1 Johdanto

Toimeksiantajana insinöörityölle oli Joiqucom, joka on työnantajani Avalonin tytäryhtiö. Tavoitteena oli tutkia ja jatko-kehittää digitaalista Joiqu-työtilaa. Tutkittiin työtilan senhetkistä toimivuutta ja toiminnallisuuksia. Päätettiin tehdä aluksi palvelun ja järjestelmän siirto toiselle palvelimelle ja palveluntarjoajalle.

Palvelulle oli tarve saada nopeutta ja vakaampi alusta, koska palvelu kärsi satunnaisesta hidastelusta. Yhteys tietokantaan oli hidas, mikä näkyi selkeästi esimerkiksi käyttöliittymän puolella, kun paljon tietoa latautui käyttöliittymään. Alustan kehitysversion nimeksi valittiin Joiqu 2.0. Tämän version tarkoituksena oli liittää Joiqu osana Visman Severa -järjestelmää. Visman puolelta oltiin kiinnostuneita tämän tyyppisestä viestintäalustasta, mikä on puuttunut heidän Severa-palvelustaan.

Suunniteltiin ja koodattiin uusi käyttöliittymä Severan ulkoasua vastaavaksi. Integraatiota varten kehiteltiin rajapinta, joka keskustelee Severan järjestelmän kanssa.

2 Digitaalinen työtila Joiqu

2.1 Taustaa

Joiqu on digitaalinen työtila, jonka tarkoituksena on, että kaikki projektit, viestintä ja tiedostot voitaisiin sijoittaa yhteen paikkaan kootusti, josta ne ovat helposti löydettävissä, seurattavissa ja kommentoitavissa.

Selainpohjainen alusta mahdollistaa ajasta ja paikasta riippumattoman käytön. Helppokäyttöinen ja mukautuva käyttöliittymä mahdollistaa myös eri päätelaitteiden käytön. Ilmoitukset sähköpostiin pitää eri keskustelijat ajan tasalla, mitä projektikeskustelussa käydään juuri sillä hetkellä. Erilliset sähköpostikeskustelut voidaan käydä täysin Joiqun digitaalisessa

viestintäalustassa. Joiqun ensimmäinen versio julkaistiin vuonna 2011 AAPO-nimellä. Nykyinen nimi vaihtui uudemman kehitysversion kautta vuoden 2016 alkuvaihteessa. Joiqua on ollut kehittämässä vuosien varrella kolme ohjelmistoyritystä. Viimeisimmän Joiqun kanssa tekemisissä olevan ohjelmistoyrityksen avainhenkilöiden vaihtaessa yritystä muodostuivat jatkokehitysasiat hankalammiksi, ja kustannukset olisivat nousseet jyrkästi. Ongelmana olivat tiedonsiirtäminen yrityksen työhenkilöiden kesken sekä Joiqun dokumentoinnin puute.

Tämän jälkeen otin Joiqun teknisenä tukena vastuulleni enemmän pienempiä teknisiä muutoksia ja korjauksia. Samassa tilanteessa tuli huomattua tutkittaessa erilaisia toiminnallisuuksia. Näistä paljastui toiminnallisuuksien puutteita, joita ei ollut viety loppuun asti.

2.2 Soveltuvuus

Digitaalinen työtila sopii erinomaisesti esimerkiksi kuntien ja julkishallinnon projekteille ja hankkeille, koska julkishallinnon työ käsittää useita organisaatioita ja sidosryhmiä. Tiedonkululle aiheutuu haasteita varsinkin suurista määristä sähköpostikeskusteluja.

Suunnittelutoimistot, asiantuntija- ja konsulttiyritykset hyötyvät Joiqun ominaisuuksista, koska työnkulku muodostuu ketterämmäksi. Tiimit ja asiakkaat voivat jakaa tiedostoja turvallisessa alustassa, keskustella sekä seurata projektien etenemistä.

2.3 Taustatekniikat

PHP

Dynaamisten verkkosivujen palvelinympäristön ohjelmointikieli. PHP on lyhenne sanoista Hypertext Preprocessor. Julkaistiin 1995 ensimmäisen kerran ja oli vuonna 2010 suosituin ohjelmointitekniikka. PHP on heikosti tyypitetty

oliopohjainen ohjelmointikieli. PHP on yleinen ohjelmointikieli muun muassa erilaisten verkkosivujen julkaisujärjestelmissä.

MySQL

MySQL on relaatiotietokantaohjelmisto, joka julkaistiin ensimmäisen kerran 1996. Nykyisin MySQL:n omistus on siirtynyt Oraclelle, joka on saatavilla vapaalla lisenssillä (GNU GLP) tai kaupallisella lisenssillä. Tietokannan päälle rakennetaan yleensä ohjelmalogiikka ohjelmointikielillä kuten Python tai PHP. Verkkosivupalvelimena yleensä toimii Apache Linuxin päällä.

JQuery

JQuery on JavaScript-kirjasto, joka on erittäin suosittu syntaksi, mikä on tehty helposti ymmärrettäväksi. Sitä käytetään muun muassa DOM-elementtien manipulointia varten. JQuerylla voidaan toteuttaa animaatioita, verkkosivujen valikoita ja muita interaktiivisia toimintoja.

HTML

Verkkosivujen rakentamisessa on käytettävä vakioitua merkintäkieltä. HTML on lyhenne sanoista Hypertext Markup Language. Merkintäkieli kertoo selaimelle verkkosivun teknisen rakenteen sekä sen, kuinka elementit tulee sijoittaa sivulle. Nykyään käytetään HTML5-versiota, joka sisältää toiminnallisuuksia muun muassa videota ja animaatioita varten.

CSS

CSS on lyhenne sanoista Cascading Style Sheets. Lyhenne tarkoittaa merkintäkieltä, jolla määritellään tyyliasetuksia muun muassa verkkosivujen elementtien tekstikokoja, värejä sekä animaatioita CSS3-versiolla.

Apache

Avoimeen lähdekoodiin kuuluvaa palvelinohjelmistoa käytetään www-palvelimilla. Yleensä PHP-pohjaisten verkkosivustojen alustana on yhdessä Linux-palvelimen päällä.

Nginx

Palvelinohjelmista käytetään www-palvelimilla. Tunnetaan palvelinratkaisuna, kun verkkosivustosta halutaan saada muun muassa parempaa suorituskykyä.

MVC

Ohjelmistoarkkitehtuuria käytetään käyttöliittymien suunnittelussa. Lyhenne tulee sanoista Model-View-Controller.

REST

REST on ohjelmointirajapintojen rakentamiseen tarkoitettu arkkitehtuurimalli. Se perustuu HTTP-tekniikkaan. REST API:n avulla voidaan tehdä integraatioita eri ohjelmistojen ja applikaatioiden välillä.

3 Suunnittelu

3.1 Tutkiminen

Projektin alussa oli tiedossa, että palvelusta ei ollut teknisiä dokumentaatioita ja määrittelyjä, joita kuuluisi olla. Palvelinalustasta oli tehty tosin kevyt kuvaus muun muassa, kuinka varmuuskopiointi ja palautus suoritetaan ja missä palvelin sijaitsee.

Dokumentoinnin puuttuminen hankaloitti alustaan tutustumista ja sen logiikan ymmärtämistä, esimerkiksi kuinka ja miten erinäiset toiminnallisuudet ja

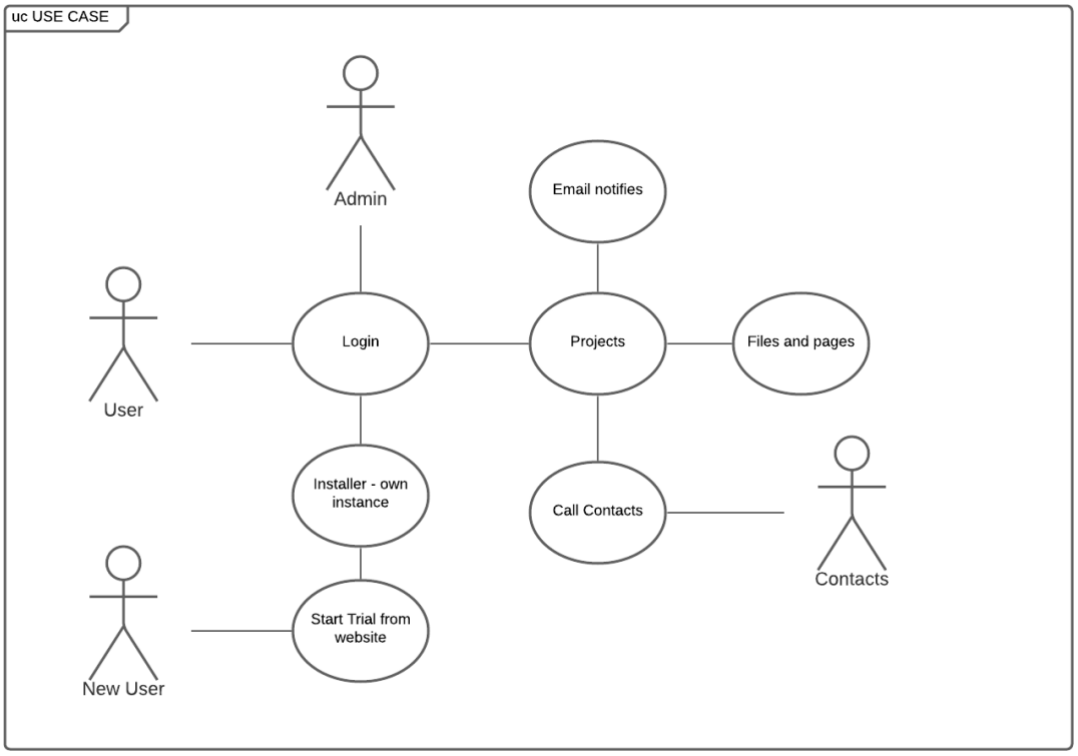
tapahtumat ovat yhteydessä toisiinsa. Voitiin myös todeta, että palvelussa oli puutteita käyttöliittymän puolella. Asiakkaiden hallinnan puolella oli esimerkiksi teknisiä ongelmia asiakastietojen käsittelyssä. Hitautta tietokantaan oli selkeästi havaittavissa ja todennettavissa niin Joiqun asiakkaiden käyttöliittymässä että asiakkaiden hallinnan käyttöliittymässä.

Arvioitiin, että nostamalla PHP-versiota palvelimella saataisiin lisää suorituskykyä. Ongelma ilmeni tässä, että tiettyjä toiminallisuuksia oli ohjelmoitu vanhalla PHP-version funktioilla, ja muutokset olisivat mahdollisesti voineet rikkoa osaltaan tai suurimmassa määrin käyttöliittymän. Olisi tarvittu kehitysalusta muutoksia ja päivityksiä varten. Korjaaminen sen hetkisellä palveluntarjoajan alustalla olisi tehnyt tämän työläämmäksi, ja kustannukset olisivat nousseet selkeästi.

Teknisiä ongelmia oli myös asiakkaan poistamisessa tai päivittämisessä. Asiakkaan poistamisen yhteydessä tiedot eivät poistuneet tietokannasta eikä poistanut tiedostoja palvelimelta vaan nuo jouduttiin poistamaan erikseen manuaalisesti. Oletettavasti toiminnallisuutta ei ole rakennettu loppuun asti. Samassa yhteydessä huomattiin lisäksi ongelmia myös muiden asiakkaiden palveluissa. Kaikki palvelut menivät alas, koska palvelin ei käynnistynyt enää. Ongelma löytyi palvelimen asetustiedostoista, joissa olivat erilliset alidomain URL -polkuviittaukset tyyliin asiakas1.joiqu.com. Jokaisella asiakkaalla on oma asiakaskohtainen alidomain ja instanssi. Alidomain rakentuu automaattisen asennusohjelman avulla, joka luo myös jokaiselle asiakkaalle oman käyttöliittymäympäristön, sivupohjat, tiedostot ja tietokannan.

Hyvällä ja selkeällä dokumentaatiolla olisivat nämä ongelma-asiat voineet olla ennakoitavissa, jolloin tutkiminen ja suunnittelu jatkokehitystä varten olisi ollut nopeampaa ja selkeämpää.

Piirsin kevyen ja yksinkertaisen use case -kuvauksen Joiqusta, koska sellainen puuttui kuten muutkin palvelun dokumentoinnit.



Kuva 1. Joiqu Use Case.

3.2 Määrittelyt

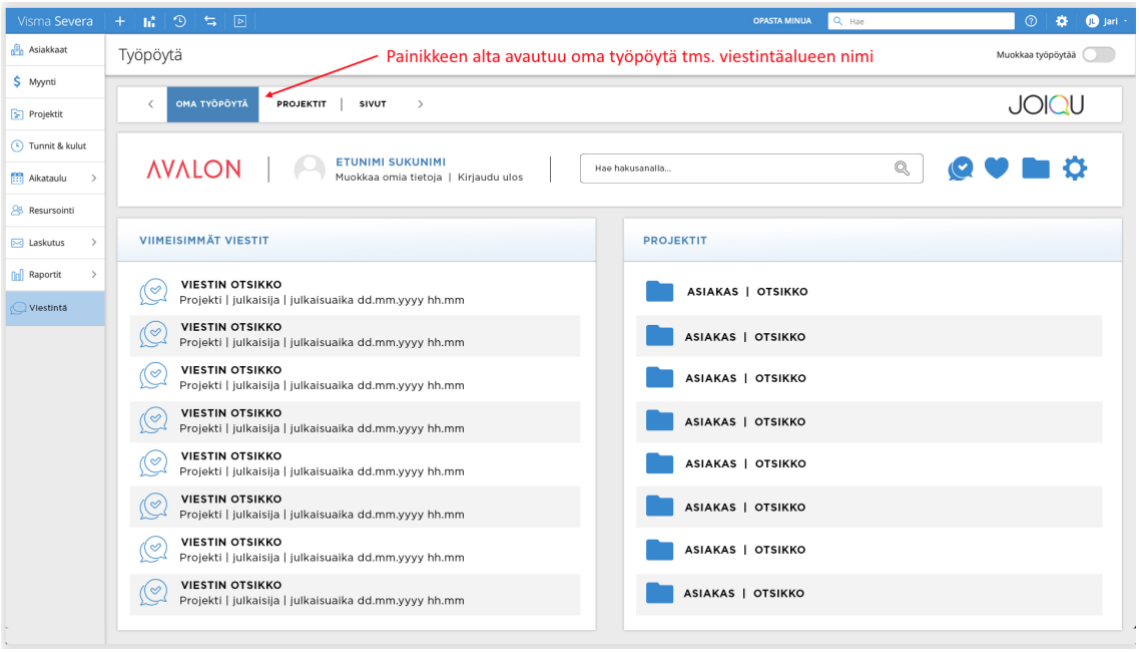
Tarpeena oli saada Joiqun asiakkaalle parempi käyttökokemus ja synergia yhdessä Visman Severan kanssa. Joiqun tarkoituksena on toimia Severan käyttäjillä digitaalisena viestintäalustana, jossa voidaan muun muassa seurata projekteihin liittyvää asiakasviestintää sekä jakaa tiedostoja ryhmien kesken. Määrittelyjä käytiin yhdessä palvelinylläpitäjämme elfGroupin kanssa. Heidän tehtävänään oli myös tehdä mahdolliset tulevat rajapintakytkennät backendissä. Joiqun tehtäväksi jäisi käyttöliittymän suunnittelu ja toteutus.

Visuaalinen suunnittelu aloitettiin ja käyttöliittymää pelkistettiin poistamalla toimintoja joita ei nähty tarpeellisiksi uudessa versiossa. Värimaailma valittiin niin ikään Severan käyttöliittymän ulkoasusta. Tarve oli saada viestitettyä asiakkaalle siitä, että molemmat käyttöliittymät kuuluvat samaan palveluun.

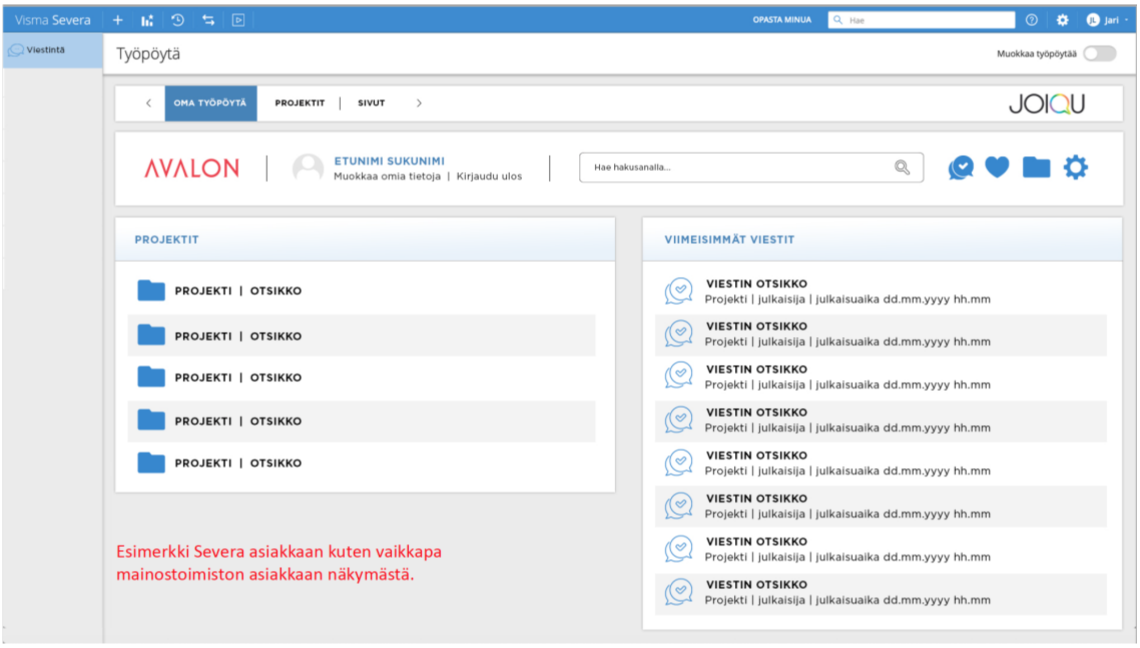
Tarpeena olivat myös kirjautumismahdollisuus ja integrointi Joiqun ja Severan välillä. Severan palvelun valinnut ottaisi Joiqun viestintäkanavat käyttöön ja pääsisi samoilla tunnuksilla siirtymälinkin kautta molempiin käyttöjärjestelmiin.

3.3 Visuaalinen suunnittelu

Suunniteltiin uusi käyttöliittymä uudella visuaalisuudella ja värimaailmalla. Visuaalinen suunnittelija eli AD hahmotteli useamman version käyttöliittymästä kuinka ja mihin kaikki elementit mahdollisesti voisivat sijoittua.



Kuva 2. Käyttöliittymän visuaalinen ulkoasu.

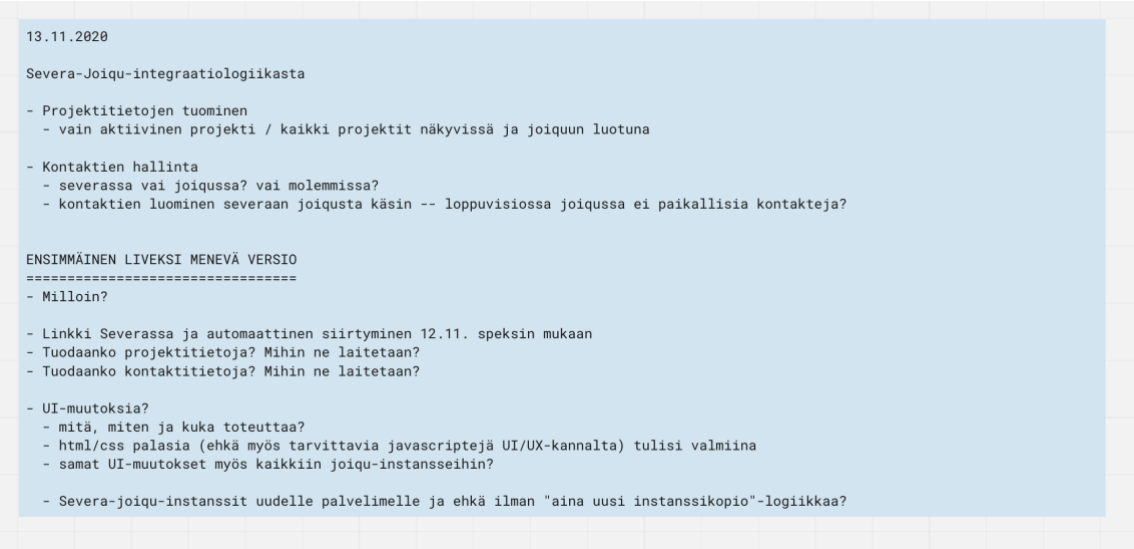


Kuva 3. Käyttöliittymän visuaalinen ulkoasu.

3.4 Integraatiosuunnittelu

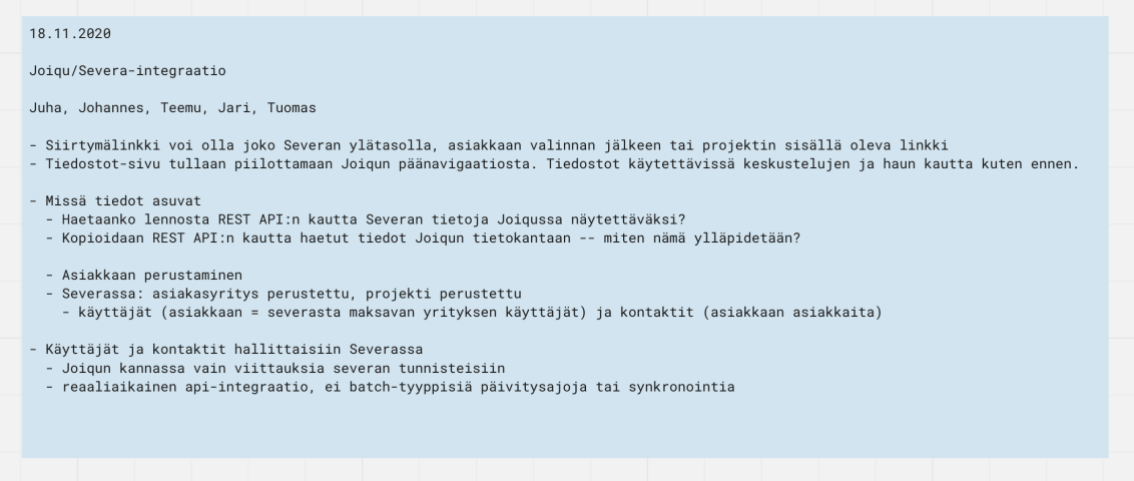
Tarpeena oli saada Joiqun integraatio toimimaan Visman Severan kanssa yhdessä samoilla Severan käyttäjätunnuksilla siirtymälinkin avulla. Alussa suunniteltiin vaihtoehtoisia toteutuksia erilaisista integraatiomalleista sekä järjestelmän jopa uudelleenrakentamisessa esimerkiksi React.js- ja Node.js-tekniikoiden ympärille.

Päädyttiin lopulta aikataulullisesti tekemään Joiqu 2.0 -versio, joka on suunniteltu varsinaisen Joiqun rinnalla toimivaksi omaksi kokonaisuudeksi. Tekniikkana taustalla toimisi edelleen PHP, MySQL ja Apache sekä Nginx. Joiqusta mietittiin myös jatkoversiota nimellä Joiqu 3.0, joka olisi juuri uudelleenrakennettu kokonaan eri tekniikoilla kuten React.js/ Vue.js, Redux, Express.js, Node.js, ja tietokantaratkaisuna toimisi MongoDB.



```
13.11.2020

Severa-Joiqu-integraatiologiikasta

- Projektitietojen tuominen
  - vain aktiivinen projekti / kaikki projektit näkyvissä ja joiquun luotuna

- Kontaktien hallinta
  - severassa vai joiqussa? vai molemmissa?
  - kontaktien luominen severaan joiqusta käsin -- loppuvisiossa joiqussa ei paikallisia kontakteja?


ENSIMMÄINEN LIVEKSI MENEVÄ VERSIO
==================================
- Milloin?

- Linkki Severassa ja automaattinen siirtyminen 12.11. speksin mukaan
- Tuodaanko projektitietoja? Mihin ne laitetaan?
- Tuodaanko kontaktitietoja? Mihin ne laitetaan?

- UI-muutoksia?
  - mitä, miten ja kuka toteuttaa?
  - html/css palasia (ehkä myös tarvittavia javascriptejä UI/UX-kannalta) tulisi valmiina
  - samat UI-muutokset myös kaikkiin joiqu-instansseihin?

  - Severa-joiqu-instanssit uudelle palvelimelle ja ehkä ilman "aina uusi instanssikopio"-logiikkaa?
```

Kuva 4. Ote muistiinpanoista 13.11. liittyen Severa-Joiqu-integraatiologiikkaan.



```
18.11.2020

Joiqu/Severa-integraatio

Juha, Johannes, Teemu, Jari, Tuomas

- Siirtymälinkki voi olla joko Severan ylätasolla, asiakkaan valinnan jälkeen tai projektin sisällä oleva linkki
- Tiedostot-sivu tullaan piilottamaan Joiqun päänavigaatiosta. Tiedostot käytettävissä keskustelujen ja haun kautta kuten ennen.

- Missä tiedot asuvat
  - Haetaanko lennosta REST API:n kautta Severan tietoja Joiqussa näytettäväksi?
  - Kopioidaan REST API:n kautta haetut tiedot Joiqun tietokantaan -- miten nämä ylläpidetään?

  - Asiakkaan perustaminen
  - Severassa: asiakasyritys perustettu, projekti perustettu
    - käyttäjät (asiakkaan = severasta maksavan yrityksen käyttäjät) ja kontaktit (asiakkaan asiakkaita)

- Käyttäjät ja kontaktit hallittaisiin Severassa
  - Joiqun kannassa vain viittauksia severan tunnisteisiin
  - reaaliaikainen api-integraatio, ei batch-tyyppisiä päivitysajoja tai synkronointia
```

Kuva 5. Ote muistiinpanoista 18.11.2020 liittyen Severa-Joiqu-integraatioon.

Todettiin, että tietokantaan tulisi rakentaa uusia tauluja integraatiota varten, tunnisteet käyttäjälle sekä projekteille ja näiden keskinäiset liitännät.

Integraatiossa valittiin rajapinnan käsittelyyn erillinen Node.js-palvelin, jonka päällä toimii Express.js-kerros, joka hoitaa parsimisen ja tiedon käsittelyn. REST API:n avulla tietoja haetaan Severasta Joiquun.

Ratkaistavana olivat muun muassa seuraavia kysymyksiä ja haasteita

- kontaktien luominen ja hallinta
- projektitietojen tuonti
- Joiqu-instanssien rakentuminen
- Severan ylätason siirtymälinkki
- REST API:n tietojen käsittely ja tallentuminen
- asiakkaan perustaminen
- integraation malli.

4 Käyttöliittymän muutokset

4.1 Sivupohjien muokkaus ja toiminnallisuudet

Käyttöliittymän koodaus ja muutokset aloitettiin kehityspalvelimella johon kopioitiin versio nykyisestä Joiqu-palvelusta. Muutoksia tehtiin sivupohjissa olevien elementtien sijoittelussa ja funktioissa.

CSS-tyylimäärityksiä muutettiin muun muassa värien, tekstin ja elementtien kokoja sekä muita asetuksia. Valikkoelementin sijoittelu sivupohjassa meni täysin uusiksi että valikko saatiin toimimaan niin pöytäversiossa sekä eri mobiililaitenäkymissä.

Käyttöliittymän ulkoasua muokattiin koodaamalla muun muassa HTML-, PHP-, CSS- ja JQuery-tekniikoilla. Sivupohjat rakentuivat useammasta erillisestä PHP-tiedostosta, joita jouduttiin muokkaamaan. Valikko oli toteutettu pääsääntöisesti JQuerylla.

Kuva 6. Kirjautumisikkuna Joiqun työtilaan vanhassa käyttöliittymässä.

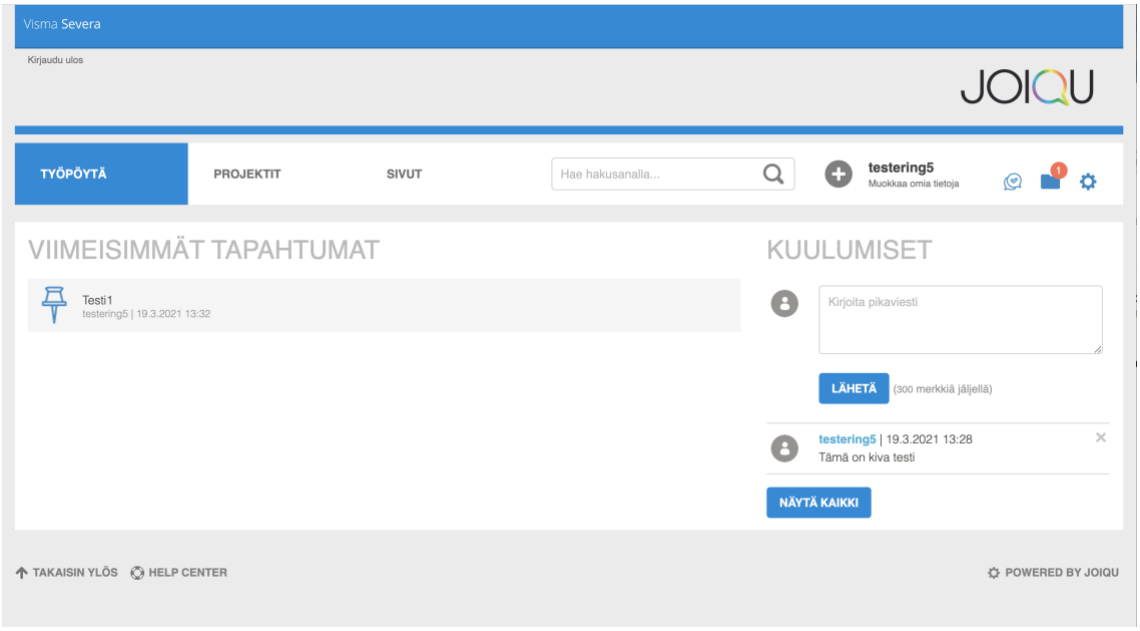
Asennusohjelma luo jokaiselle asiakkaalle oman käyttöliittymäympäristönsä. Asennusohjelma käynnistyy automaattisesti esimerkiksi tilaamalla kokeiluversion Joiqun verkkosivujen kautta. Lomakkeen lähettämisen jälkeen järjestelmä lähettää tiedot tunnuksista, joiden avulla voidaan kirjautua käyttöliittymän ympäristöön.

Kuva 7. Vanhan käyttöliittymän ulkoasu.

Uuden käyttöliittymän kirjautumisikkunaan ei tehty juurikaan muutoksia lukuun ottamatta kirjaudu-painikkeen sinistä väriä sekä sivun taustavärin vaihtaminen eri harmaan sävyiseksi.

Varsinaisen uuden käyttöliittymän ulkoasu muokattiin ulkoasusuunnitelman pohjalta. Jouduttiin hieman koodauksen edetessä poikkemaan alustavasta ulkoasusta, koska välttämättä kaikkia toiminallisuuksia ja muutoksia ei ollut vielä aivan suunniteltu loppuun asti. Valikko siirrettiin vasemmalta käyttöliittymään vaakatasoon eli horisontaaliseksi. Valikko muuttuu selaimen näkymässä eri mobiililaitteissa siten, että valikon saa esille erillisellä painikkeella käyttöliittymän oikeasta yläkulmasta. Tätä kutsutaan joissain yhteyksissä myös ”hampurilaisvalikoksi”.

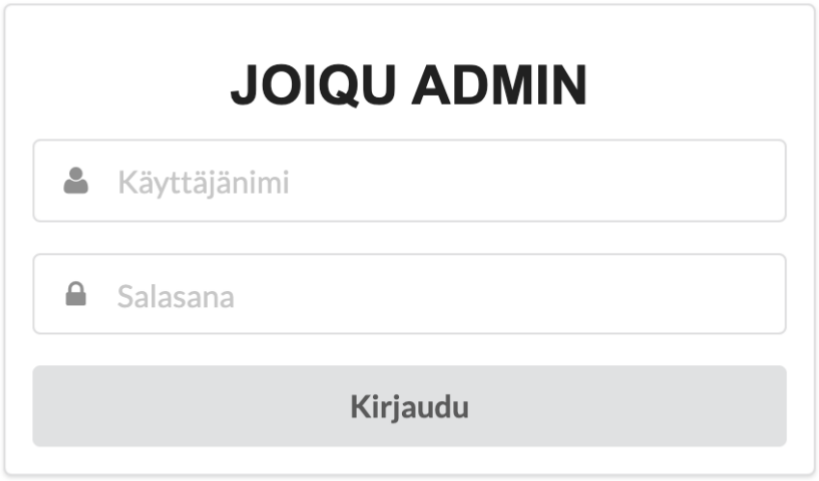
Käyttöliittymän vasemmassa yläosasta on suora linkki Visman Severaan. Samaisessa sinisessä yläpalkissa on tilaa mahdollisesti siihen sijoitettaviin lisälinkkeihin, mikäli integraatio mahdollistaisi toiminnot.



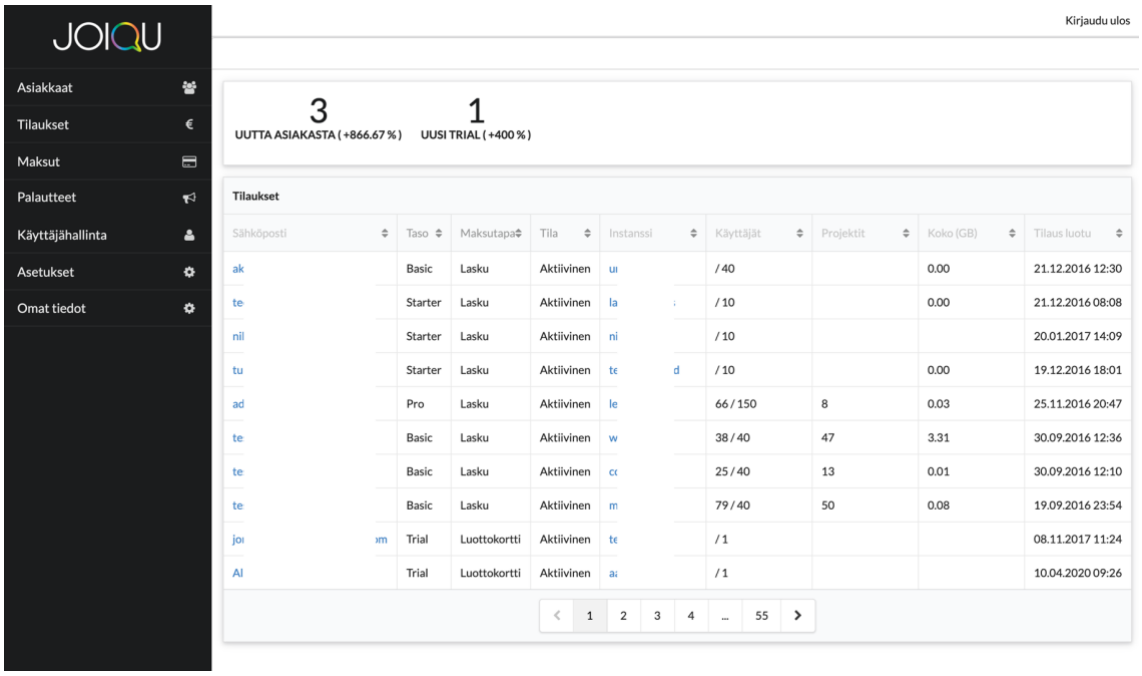
Kuva 8. Uuden käyttöliittymän testiversio kokeilussa.

Joiqun asiakashallinnan käyttöliittymä jäi tässä suunnittelun vaiheessa muuttamatta, vaikkakin tiedettiin sielläkin erinäisiä parannus- ja kehityskohteita.

Nämä parannukset tullaan toteuttamaan seuraavissa kehitysvaiheissa. Tarpeina ovat muun muassa parempi raportointi, laskutus ja asiakashallinta.



Kuva 9. Kirjautumisikkuna Joiqu backend.



Kuva 10. Joiqun backend-asiakashallinta.

4.2 Integraation valmistelut

Tiedettiin, että Joiqussa on käytetty aikaisemmin integraatioita muun muassa ValueFramen kanssa. Käyttäjät ja salasanat tarkistettiin rajapinnan kautta ValueFramessa. Olin poistanut aikaisemmin muutamasta asiakuudesta jo tarpeettoman integraatio kytkennän. ValueFramessa integraatiota varten oli tehty aikaisemmin noin 37 varsinaista model-tiedostoa sekä omat config-tiedostot.

4.3 Integraatio

Ensimmäiseen integraatiovaiheeseen testausta varten tarvittiin tehdä tietokannan tauluihin uudet kenttätiedot. Tarkasteltiin tietokannan rakennetta tiettyjen taulujen osalta kuten users, customer, projects. Voitaisiinko noihin tauluihin rakentaa vastineet Severaa varten muun muassa severa_project_id ja severa _user_id kentät.

Tarkasteltiin ja vertailtiin kenttätietoja vanhasta ValueFrame-integraatiosta. Customer-taulussa oli tehty ValueFramea varten tunniste-vfid tietotyyppinä int.

vfid = ValueFrame ID vanhasta integraatiosta ...

```
mysql> describe customer;
+------------+--------------+------+-----+-------------------+----------------+
| Field      | Type         | Null | Key | Default           | Extra          |
+------------+--------------+------+-----+-------------------+----------------+
| id         | int(11)      | NO   | PRI | NULL              | auto_increment |
| name       | varchar(255) | YES  |     | NULL              |                |
| department | varchar(255) | YES  |     | NULL              |                |
| address    | varchar(255) | YES  |     | NULL              |                |
| postalcode | varchar(45)  | YES  |     | NULL              |                |
| city       | varchar(45)  | YES  |     | NULL              |                |
| logo       | int(11)      | YES  |     | NULL              |                |
| theme      | int(11)      | YES  |     | NULL              |                |
| added      | timestamp    | YES  |     | NULL              |                |
| updated    | timestamp    | YES  |     | CURRENT_TIMESTAMP |                |
| updated_by | int(11)      | YES  |     | NULL              |                |
| added_by   | int(11)      | YES  |     | NULL              |                |
| vfid       | int(11)      | YES  |     | NULL              |                |
| folder_id  | int(11)      | YES  |     | NULL              |                |
+------------+--------------+------+-----+-------------------+----------------+
14 rows in set (0.00 sec)
```

Kuva 11. ValueFrame ID vanhassa integraatiossa.

```
mysql> describe project;
+-------------+--------------+------+-----+-------------------+----------------+
| Field       | Type         | Null | Key | Default           | Extra          |
+-------------+--------------+------+-----+-------------------+----------------+
| id          | int(11)      | NO   | PRI | NULL              | auto_increment |
| name        | varchar(255) | YES  |     | NULL              |                |
| description | text         | YES  |     | NULL              |                |
| content     | text         | YES  |     | NULL              |                |
| customer_id | int(11)      | YES  |     | NULL              |                |
| parent_id   | int(11)      | YES  | MUL | NULL              |                |
| created     | timestamp    | YES  |     | NULL              |                |
| status      | smallint(6)  | NO   |     | 0                 |                |
| updated     | timestamp    | YES  |     | CURRENT_TIMESTAMP |                |
| vfid        | int(11)      | YES  |     | NULL              |                |
+-------------+--------------+------+-----+-------------------+----------------+
10 rows in set (0.00 sec)
```

Kuva 12. Projektitaulun rakenne.

```
mysql> describe users;
+---------------------------+--------------+------+-----+-------------------+----------------+
| Field                     | Type         | Null | Key | Default           | Extra          |
+---------------------------+--------------+------+-----+-------------------+----------------+
| id                        | int(11)      | NO   | PRI | NULL              | auto_increment |
| vfid                      | int(11)      | YES  |     | NULL              |                |
| username                  | varchar(64)  | NO   | UNI | NULL              |                |
| name                      | varchar(128) | YES  |     | NULL              |                |
| password                  | varchar(128) | YES  |     | NULL              |                |
| vfsession                 | varchar(64)  | YES  |     | NULL              |                |
| created                   | timestamp    | NO   |     | CURRENT_TIMESTAMP |                |
| avatar                    | varchar(128) | YES  |     | NULL              |                |
| internal                  | tinyint(1)   | YES  |     | 1                 |                |
| customer_id               | int(11)      | YES  |     | NULL              |                |
| vfcustomerid              | int(11)      | YES  |     | NULL              |                |
| vfcontactid               | int(11)      | YES  |     | NULL              |                |
| language                  | varchar(3)   | YES  |     | NULL              |                |
| theme                     | varchar(128) | YES  |     | NULL              |                |
| email                     | varchar(128) | YES  |     | NULL              |                |
| logo                      | varchar(512) | YES  |     | NULL              |                |
| active                    | tinyint(1)   | NO   |     | 1                 |                |
| comment_notifications     | tinyint(1)   | NO   |     | 0                 |                |
| daily_digest              | tinyint(1)   | NO   |     | 0                 |                |
| bg_img                    | varchar(128) | YES  |     | NULL              |                |
| phone                     | varchar(128) | YES  |     | NULL              |                |
| title                     | varchar(125) | YES  |     | NULL              |                |
| company                   | varchar(255) | YES  |     | NULL              |                |
| address                   | varchar(255) | YES  |     | NULL              |                |
| readonly                  | tinyint(1)   | YES  |     | 0                 |                |
| locked                    | tinyint(1)   | YES  |     | 0                 |                |
| token                     | varchar(255) | YES  | UNI | NULL              |                |
| vfpassword                | varchar(128) | YES  |     | NULL              |                |
| city                      | varchar(128) | YES  |     | NULL              |                |
| postalcode                | varchar(6)   | YES  |     | NULL              |                |
| skype                     | varchar(128) | YES  |     | NULL              |                |
| linkedin                  | varchar(128) | YES  |     | NULL              |                |
| seen_in_aapo              | tinyint(1)   | YES  |     | 0                 |                |
| deleted                   | tinyint(1)   | YES  |     | 0                 |                |
| acc_details_emailed       | timestamp    | YES  |     | NULL              |                |
| comment_notifications_sms | tinyint(1)   | YES  |     | 1                 |                |
| timezone                  | varchar(100) | YES  |     | NULL              |                |
+---------------------------+--------------+------+-----+-------------------+----------------+
37 rows in set (0.01 sec)
```

Kuva 13. Käyttäjät eli users-taulun rakenne.

Tietokantaan tarvittiin uudet tunnisteet, joihin voitaisiin rajapinnan kautta siirtää asiakkaiden tiedot sekä projektitiedot Severasta Joiquun. Tarvittiin kenttätiedot muun muassa projekteille ja niiden liitännät kontakteille, jotka on liitetty kyseisiin projekteihin.

```
mysql> select * from project where id>3;
+----+---------------------+-------------+---------+-------------+-----------+---------+--------+---------------------+------+--------------+
| id | name                | description | content | customer_id | parent_id | created | status | updated             | vfid | severa_guid  |
+----+---------------------+-------------+---------+-------------+-----------+---------+--------+---------------------+------+--------------+
|  7 | elfgroupin projekti | NULL        | NULL    |           6 |      NULL | NULL    |      0 | 2021-03-17 22:08:46 | NULL | [redacted]   |
|  8 | Integraatioprojekti | NULL        | NULL    |           7 |      NULL | NULL    |      0 | 2021-03-17 22:08:46 | NULL | [redacted]   |
|  9 | Oma sisäinen projekti | NULL      | NULL    |           7 |      NULL | NULL    |      0 | 2021-03-17 22:08:46 | NULL | [redacted]   |
+----+---------------------+-------------+---------+-------------+-----------+---------+--------+---------------------+------+--------------+
3 rows in set (0.00 sec)
```

Kuva 14. Projektien taulukosta testausvaiheessa.

Luotiin lopuksi Severaa varten tietokantaan testausta varten erikseen oma config-taulu, jonne tallentuvat Severan API-tunnisteet. Määriteltiin taulukon name-kentän tietotyypille varchar-koon arvoksi 128 ja value-kentälle tarvittavat 2048.

```
mysql> create table config (name varchar(128), value varchar(2048));
Query OK, 0 rows affected (0.02 sec)

mysql> insert into config (name, value) values ('severa_api_client_id', '                              .apps.vismasevera.com');
Query OK, 1 row affected (0.00 sec)

mysql> insert into config (name, value) values ('severa_api_client_secret', '                                        ');
Query OK, 1 row affected (0.00 sec)

mysql> select * from config;
+--------------------------+--------------------------------------------------+
| name                     | value                                            |
+--------------------------+--------------------------------------------------+
| severa_api_client_id     |                              .apps.vismasevera.com |
| severa_api_client_secret |                                                   |
+--------------------------+--------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

Kuva 15. Kuva testausvaiheesta tietokannan config-taulusta.

Käyttöliittymään valmisteltiin erikseen paikka, josta voidaan tallentaa Severan Client ID sekä Severan Client Secret -tunnisteet config-tauluun. Tunnisteet ovat Joiqun asiakaskohtaisia, joten tiedot kuten projektit välittyvät rajapinnan kautta Severasta vain kyseiseen asiakuuteen. Tarvitaan myös erikseen severa_customer-tyyppinen taulu, johon voidaan tallentaa Severan asiakuuteen liittyviä tietoja.

```
68  async function getCustomerFromProjectTableBySeveraGUID(severa_guid) {
69
70    const [rows, fields] = await con.execute("SELECT id from project where parent_id is null and severa_guid = ?", [severa_guid])
71
72    for (let i=0; i < rows.length; i++) {
73      return rows[i].id
74    }
75
76    return null
77  }
78
79  async function insertCustomerIntoProjectTable(name, customer_id, severa_guid) {
80
81    var sql = "INSERT INTO project (name, description, customer_id, parent_id, status, severa_guid) VALUES (?,?,?,?,?,?)"
82    const res = await con.execute(sql, [name, name, customer_id, null, 1, severa_guid])
83    return res
84  }
85
```

Kuva 16. Kuvausta rajapinnasta.

```
async function getProjects() {
  try {
    let config = {
      headers: {
        "Client_id": VISMA_CLIENT_ID,
        "Authorization": "Bearer " + access_token
      }
    }

    let res = await axios.get( url: URL_VISMA_API+'projects', config)

    let takethis=false
    for (let i=0; i < res.data.length ; i++) {
      // console.log("received proj: " + JSON.stringify(res.data[i]))
      let guid = res.data[i].guid
      let name = res.data[i].name
      let customer_guid = res.data[i].customer.guid

      let customer_id = await getCustomerIdBySeveraGUID(customer_guid)

      let existingProjId = await getProjectIdBySeveraGUID(guid)
      if (existingProjId) {
        console.log("Project " + existingProjId + " already in DB")
        continue
      }

      let custInProj = await getCustomerFromProjectTableBySeveraGUID(customer_guid)
      if (custInProj == null) {
        await insertCustomerIntoProjectTable(res.data[i].customer.name, customer_id, customer_guid)
        custInProj = await getCustomerFromProjectTableBySeveraGUID(customer_guid)
      }

      console.log("Inserting project " + guid + " / " + name + " / " + customer_id + " / " + customer_guid + " / " +custInProj)

      var sql = "INSERT INTO project (name, description, customer_id, parent_id, status, severa_guid) VALUES (?,?,?,?,?,?)"

      con.query(sql, [name, name, null, custInProj, 1, guid], function (err, result) {
        if (err) throw err;
        console.log("Project record inserted");
      })

    }

  } catch (error) {
    console.error(error)
  }

}
```

Kuva 17. Kuvausta rajapinnasta. Projektien haku.

5 Haasteet, jatkokehitystyö ja lopputulema

Projektin aikana haasteena on ollut muun muassa aikataulutus, koska olen joutunut rakentamaan kehitysprojektia muiden asiakastöiden ohessa selkeästi sivuprojektina. Näin ollen projekti ei ole edennyt niin joutuisasti kuin olisi pitänyt.

Joiqusta puuttuvat alkuperäiset tekniset dokumentoinnit hidastivat varsinkin projektin alkuvaiheessa huomattavasti. Hyvän dokumentoinnin ansioista olisi voitu hahmottaa ohjelmiston logiikkaa ja se, miten kaikki toiminnallisuudet ovat yhteydessä toisiinsa. Koodien joukosta puuttuivat suurimmalta osin

kommentoinnit tai selitteet, joita koodien funktiossa tapahtuu ja mihin muihin tiedostoihin sillä on vaikutusta.

Projektin edetessä myös integraation kohdalla ilmeni viivästyksiä ja varsinkin varsinainen yllätys viime hetkellä, kun kyseisellä siirtymälinkkiintegraatio tavalla, joka alun perin oli sovittu, ei kuitenkaan voitu sellaisenaan toteuttaa. Siirtymälinkin kautta siirtymisen hylkäämistä Joiquun oli perusteltu muun muassa tietoturva-asioilla, vaikkakin siirtymälinkin käyttäminen ja tietoturvallisuus oli tarkistettu jo aikaisemmin kyberturvallisuusyritys elfGroupin taholta.

Asiakkaat ja käyttäjät joudutaan jatkossakin luomaan edelleen molemmissa käyttöjärjestelmissä, niin Joiqussa että Severassa. Projektit siirtyvät REST API:n kautta Joiquun.

Joiqun osalta suunnittelu jatkuu vielä tämän insinöörityön aikana ja jälkeenkin eri kehitysversioissa, joten olemme kartoittaneet myös yhteistyökumppanimme elfGroupin kanssa toteuttajaa, joka voisi jatkaa projektia eteenpäin. Työtilanteeni ei mahdollista tällä hetkellä täysipainoista keskittymistä kyseiseen Joiqu-Severa-uudistusprojektiin muiden asiakasprojektien lomassa.

Joiqusta tuleva uusi kaavailtu kehitysversio 3.0 tarvitsee täydellisen tekniikan muutoksen niin koodissa, sekä tietokanta- että palvelintasolla. Nähtäväksi vielä jää, olenko mukana uudemman version suunnittelu- tai tuotantovaiheessa. Tarkoituksena on tehdä vielä projektin edetessä dokumentoinnit tai lisäpiirrokset kuten muun muassa activity- ja class diagramit. Kevyt use case-kuvaus tehtiin projektin edetessä.

Tämän insinöörityön tekeminen ja oleminen osana Joiqun kehitystyötä on ollut hyvin opettavainen matka ohjelmistokehitystyöhön. Erinäiset aikataululliset ja tekniset haasteet, suunnitteluvaiheet on pyritty ottamaan huomioon prosessin eri osa-alueissa ja vaiheissa. Yllätyksiltä ei voitu välttyä tässä projektissa varsinkin, kun se liittyy alun perin suunniteltuun integraatiotapaan kuten siirtymälinkin käyttöön. Tämä oli alussa selkeää ja annettu ymmärtää, että

voisimme edetä ja toteuttaa tämän integraation siirtymälinkin avulla Severasta Joiquun. Erinäisten haasteiden ja teknisten kokonaisuuksien hyvä hahmottaminen on erityisen tärkeää ohjelmistosuunnittelun kokonaisprosessissa.

6 Yhteenveto

Alussa Joiqun palvelut siirrettiin toiselle palvelimelle ja palveluntarjoajalle. Näin saatiin vakautta ja nopeutta palvelulle. Joiqun kehitysprojekti käynnistettiin alkuun tarvekartoituksella ja suunnittelulla. Alussa käytiin muutamia kehityspalavereita niin suunnittelijoiden kuin myynninkin kanssa.

Hahmoteltiin myös aikataulutusta toiminallisuuksia, integraatiomahdollisuuksia sekä ulkoasua. Aloitettiin käyttöliittymän koodaus uusiksi, käyttöliittymän elementtien uudelleensijoittelut sekä tarpeettomien toiminallisuuksien poistaminen. Tietokantaan rakennettiin lisäkenttiä tauluihin kuten muun muassa users, customer ja project. Tarvittiin integraatio Visman Severan kanssa siirtymislinkin kautta. Kun siirtymälinkkiä ei voitu toteuttaa, tehtiin aluksi kovakoodattu manuaalinen integraatio testaustarkoituksessa.

Tarvittiin käyttöliittymään paikka, josta voidaan tallentaa Visman Severan API key ja API secret erilliseen tietokannassa olevaan config-tauluun. Lopuksi päädyttiin käyttämään muun muassa Node.js-palvelinta yhdessä Express.js:n kanssa, jolloin integraatio toteutettiin REST API:n avulla Express.js:n toimiessa tiedon parsijana ja välittäjänä.

Projektin edetessä todettiin, että tarvitaan erikseen jatkokehitystä varten mahdollisesti elfGroupin kautta ohjelmistoyritys yhteistyökumppaniksi, joka voisi tehdä kehitystyötä ja viedä eteenpäin projektia.

Lähteet

Digitaalinen työtila Joiqu. Verkkoaineisto. <https://www.joiqu.com/>. Luettu 26.12.2020.

PHP. Dokumentaatiota ohjelmointikielestä. Verkkoaineisto. <https://www.php.net/>. Luettu 7.11.2020.

MySQL. Dokumentaatiota tietokannasta. Verkkoaineisto. <https://www.mysql.com/>. Luettu 7.11.2020.

Apache. Dokumentaatiota palvelinohjelmasta. Verkkoaineisto. <https://httpd.apache.org/>. Luettu 7.11.2020.

Nginx. Dokumentaatiota palvelinohjelmasta. Verkkoaineisto. <https://www.nginx.com/>. Luettu 7.11.2020.

Node.js. Dokumentaatiota palvelinratkaisusta. Verkkoaineisto. <https://nodejs.org/en/>. Luettu 25.11.2020.

Express.js. Dokumentaatiota verkkosovelluskehyksestä. Verkkoaineisto. <https://expressjs.com/>. Luettu 25.11.2020.

React.js. Dokumentaatiota JavaScript-kirjastosta. Verkkoaineisto <https://reactjs.org/>. Luettu 25.11.2020.

MVC Framework Tutorial. Verkkodokumentti. <https://www.tutorialspoint.com/mvc_framework/index.htm>. 26.12.2020.

MongoDB. Dokumentointia tietokannasta. Verkkoaineisto. <https://www.mongodb.com/>. Luettu 25.11.2020.

Visma Severa Rest API. Rajapinnan dokumentaatio. Verkkoaineisto. <https://api.severa.visma.com/rest-api/doc/index.html?urls.primaryName=Version%200.2>. Luettu 26.1.2021.