



# Ulosottojärjestelmän käyttöliittymien toteutus moderneilla verkkokehitysteknologioilla

Teemu Tannerma

Toni Vääntinen

OPINNÄYTETYÖ  
Marraskuu 2020

Tietojenkäsittely  
Web-palvelut ja Ohjelmistotuotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittely  
Web-palvelut ja ohjelmistotuotanto

TANNERMA, TEEMU & VÄNTTINEN TONI:

Ulosottojärjestelmän käyttöliittymien toteutus moderneilla verkkokehitysteknologioilla

Opinnäytetyö 38 sivua, joista liitteitä 0 sivua  
Marraskuu 2020

---

Opinnäytetyön tarkoituksena oli toteuttaa ulosottojärjestelmälle kaksi eri käyttöliittymää: Management-käyttöliittymä ja Agent-käyttöliittymä, joka on tarkoitettu myyntiagenteille. Opinnäytetyössä dokumentoitiin sovelluksien suunnittelua, kehitystyötä ja toiminnallisuutta.

Verkkosovelluksien kehityksessä käytettiin moderneja verkkokehitysteknologioita kuten React-kirjastoa ja Redux-tilanhallintakirjastoa. Verkkosovelluksien pohja luotiin create-react-app-työkalulla. Verkkosovelluksien käyttämä palvelinrajapinta oli .NET Core -pohjainen REST-rajapinta. Opinnäytetyössä paneudutaan aluksi projektissa käytettyihin verkkokehitysteknologioihin ja niiden käyttötarkoitukseen. Tämän jälkeen raportissa käsitellään projektiin liittyvää suunnittelua ja sitä, millaisia vaatimuksia asiakkaalla oli käyttöliittymien osalta. Raportissa arvioidaan saatuja tuloksia ja selvitetään asiakkaan vaatimusten täyttymistä. Raportin lopuksi selvitetään, mitä projektissa olisi voinut tehdä paremmin ja mitkä asiat onnistuivat.

Molemmat verkkosovellukset saatiin valmiiksi aikataulun mukaisesti, vaikka alkuperäistä julkaisupäivämäärää siirrettiin noin neljä kuukautta aikaisemmaksi. Kehitystyö ja käyttöliittymien testaus vaati suuria ponnistuksia kehittäjiltä. Verkkosovellukset täyttivät asiakkaan tärkeimmät vaatimukset. Asiakas oli myös innokas jatkamaan tuotteen kehitystä, ja heillä oli useita uusia kehitystöitä ajateltuna valmiiksi.

---

Asiasanat: verkkokäyttöliittymä, agent-käyttöliittymä, management-käyttöliittymä, ulosottojärjestelmä, react, redux, create-react-app

## **ABSTRACT**

Tampere University of Applied Sciences  
Degree Programme in Business Information Systems  
Web Services Software Production

TANNERMA, TEEMU & VÄNTTINEN TONI:  
Development of Outcall Service's User Interfaces with Modern Web Development Technologies

Bachelor's thesis 38 pages, appendices 0 pages  
November 2020

---

The aim of the thesis was to implement two different user interfaces for a call-out system: Management user interface and Agent user interface for sales agents. This thesis documents the design, development work, and functionality of the applications.

Modern web development technologies, such as React and Redux state management library, were used in the development of the web applications. The web application template was created with create-react-app tool. The server interface used by web applications was a .NET Core-based REST API. The thesis first focuses on the web development technologies used in the project and their purpose. The report then discusses the design associated with the project and what demands the customer had in terms of user interfaces. The report evaluates the results obtained and finds out whether the customer's requirements have been met. At the end of the report, there is a review of what could have been done better in the project and what aspects were successful.

Both web applications were completed on schedule, although the original release date was moved to be approximately four months earlier. The development work required much effort from the developers, and there had to be plenty of testing for the user interfaces. The web applications met the customer's most important requirements. The customer was also eager to continue product development and had several new development ideas at the end.

---

Key words: web interface, agent interface, management interface, callout system, react, redux, create-react-app

## SISÄLLYS

1	JOHDANTO .....	8
2	JAVASCRIPT .....	9
	2.1 TypeScript.....	9
	2.2 TypeScript ja React.....	10
3	REACT .....	11
	3.1 JSX .....	11
	3.2 Komponentit .....	11
	3.3 React Hooks .....	12
	3.4 Redux.....	13
4	TARVITTAVAT OMINAISUUDET JA SUUNNITTELU .....	15
	4.1 Management-käyttöliittymä .....	15
	4.2 Agent-käyttöliittymä.....	16
5	MANAGEMENT-NÄKYMÄN TOTEUTUS .....	18
	5.1 Kirjautuminen .....	18
	5.2 Navigaatio .....	19
	5.3 Kampanjat-näkymä .....	19
	5.4 Yksittäinen kampanjanäkymä .....	20
	5.4.1 Kampanjan asetukset .....	21
	5.4.2 Kontaktit .....	25
	5.4.3 Tuotteet .....	26
	5.4.4 Kysymykset .....	27
	5.5 Raporttinäkymä .....	27
	5.6 Kalenteri-näkymä .....	28
6	AGENT-NÄKYMÄN TOTEUTUS .....	29
	6.1 Kirjautuminen .....	30
	6.2 Kontakti-sivu .....	30
	6.3 Tuotteet ja kysymykset -sivu .....	31
	6.4 Lopputuloksen kirjaaminen .....	33
	6.5 Kalenteri-sivu .....	34
7	TULOS .....	35
	7.1 Management-käyttöliittymä .....	35
	7.2 Agent-näkymä .....	35
8	POHDINTA .....	37
	LÄHTEET .....	38

**LYHENTEET JA TERMIT**

Outbound-markkinointi	Outbound-markkinointi on markkinointitapa. Siinä markkinoija etsii ja lähestyy itse potentiaalisia asiakkaita. Yksi outbound-markkinoinnin kanava on telemarkkinointi. (Vähä-Ruka, 2015).
Telemarkkinointi	Suoramarkkinointia tuotteiden tai palveluiden potentiaalisille asiakkaille puhelimitse, Internetissä tai faxilla. Telemarkkinointi on outbound-markkinointia. (Kenton, 2020).
Puhelinpalvelukeskus	Puhelinpalvelukeskus eli Call Center on toimipiste, joka tuottaa puhelinpalveluita.
Myyntikampanja	Suunniteltu myyntistrategia, jolla hankitaan asiakkaita. Myyntikampanjalle on määritelty myytävät tuotteet sekä erikseen myyntitilanteessa asiakkaalta kysyttävät kysymykset.
Management UI	Ulosottojärjestelmän hallintapaneeli, josta voi hallita myyntikampanjoita, tulostaa raportteja, tai tarkastella ja muokata työntekijöiden työaikoja. Tästä hallintapaneelistä voi lisätä esimerkiksi kysymyksiä ja tuotteita, joita myyntiagentit näkevät Agent-käyttöliittymästä.
Agent-käyttöliittymä	Myyntiagenttien käyttöliittymä, joka on integroitu ulosottojärjestelmään. Agent-käyttöliittymästä myyntiagentit näkevät soitetun asiakkaan tiedot sekä myyntikampanjaan kuuluvat tuotteet ja kysymykset.
ES5, ES6	ECMAScript-standardin eri versioita, joilla varmistetaan yhteentoimivuus verkkoselainten välillä. ECMAScript-

standardeja voidaan myös ilmaista vuosiluvuilla kuten ES2016. (Flanagan, 2020, 1).

Superset	Paranneltu ohjelmointikieli. Tarjoaa kaikki ominaisuudet tietystä ohjelmointikielestä, sekä laajentaa tai parantaa kyseistä ohjelmointikieltä uusilla ominaisuuksilla.
SPA	SPA eli single-page-application on verkkosovellus tai verkkosivu, joka sisältää vain yhden sivun. Sivun näkyvä päivitys hakemalla dataa palvelimelta ja päivittämällä sivulla näkyvät tiedot.
ReactJS	ReactJS on Facebookin kehittämä avoimen lähdekoodin käyttöliittymäkomponenttikirjasto. Käyttöliittymäkomponentit luodaan Reactin avulla käyttäen JavaScriptiä.
JSX	JSX on JavaScript-lisäosa, missä on XML:n kaltainen syntaksi. Se yhdistää HTML:n ja JavaScriptin, jonka avulla Reactissa voidaan luoda elementtejä helposti.
HTML	HTML eli Hypertext Markup Language. HTML on standardi merkintäkieli verkkosivuille.
CSS	CSS eli Cascading Style Sheets on kieli, jolla tyylitellään HTML-dokumentti. CSS:n avulla voidaan määrittellä se, miten HTML-elementit näytetään verkkosivulla.
PaaS	PaaS eli Platform as a Service on alustapohjainen palvelukategoria. PaaS:n periaate on, että palveluntarjoaja toimittaa työkalun tai ohjelmiston käyttäjille internetin välityksellä.

Token

Digitaalinen tunniste, jota käytetään token-pohjaisessa tunnistautumisessa, jotta sovellus voi ottaa yhteyden ohjelmointirajapintaan.

## 1 JOHDANTO

Telemarkkinointi on yksi johtavista outbound-markkinoinnin muodoista. Telemarkkinointia suoritetaan puhelinpalvelukeskuksista, joissa työskentelee useampi myyntiagentti. Nämä puhelinpalvelukeskukset tarvitsevat ohjelmistoratkaisun, joilla he voivat soittaa asiakkaille, seurata agenttien myyntiä, suorittaa työajanhallintaa, sekä tulostaa raportteja kampanjoista. Puhelinpalvelukeskusten lisääntymisen myötä tarve call center -ohjelmistoratkaisuille on kasvanut.

Tässä parityönä suoritettussa opinnäytetyössä dokumentoimme call center -verkkosovelluksien kehitysprosessia ja sen ominaisuuksia. Toimeksiantoon kuului kehittää hallintapaneeli, jossa voi hallita eri myyntikampanjoita, työntekijöiden työaikaa sekä tulostaa myyntiraportteja. Myyntiagenteille tuli kehittää oma sovellus. Myyntiagenttien sovelluksesta tuli nähdä myyntitilanteessa vaaditut tiedot asiakkaasta, tuotteista ja mahdollisista kysymyksistä.

Verkkosovelluksen käyttöliittymä kehitettiin TypeScript-ohjelmointikielellä ja siihen liittyvällä React-kirjastolla. Tyylikirjastona käytettiin Material UI:ta. Verkkosovellus on yhden sivun sovellus. REST-rajapinta, jota verkkosovellus käyttää tietojen hakemisessa ja päivittämisessä, on kehitetty ASP.NET Core -ohjelmistokehyksellä. Rajapinta on yrityksemme itse kehittämä, ja se on kustomoitu verkkosovellustamme varten.

Opinnäytetyö on dokumentaatio kahden eri verkkosovelluksen kehitysprosessista ja suunnittelusta. Dokumentoimme myös sovelluksien toimintaa yksityiskohtaisesti. Sovelluksien käyttämää REST-rajapintaa dokumentoimme pintapuolisesti. Käymme läpi, miten valitsemamme teknologiat soveltuivat verkkosovelluksen kehittämiseen. Opinnäytetyö on ensisijaisesti suunnattu yrityksellemme (Visma AI Works) dokumentaatioksi. Opinnäytetyöstä on apua kehittäjille, jotka haluavat ymmärtää nykyaikaisia verkkosovellusteknologioita ja miten niitä käytetään modernissa ohjelmistoyrityksessä. Yritys voi käyttää opinnäytetyötä viitteenä, jos he haluavat tehdä muutoksia verkkosovelluksiin, tai kun he ovat kehittämässä uutta verkkosovellusta.

## 2 JAVASCRIPT

JavaScript on järjestelmäriippumaton oliopohjainen skriptauskieli, jolla voidaan luoda interaktiivisia verkkosivuja. JavaScriptin avulla verkkosivu voi tehdä rajapintakutsuja tai palvelinkutsuja ilman, että verkkosivua tarvitsisi jokaisen kutsuntakia ladata uusiksi. Tämä vähentää verkkoliikennettä ja tekee sivuista nopeampia käyttää. Verkkosivun rakennetta ja tyyliä voidaan hallita tekemällä muokkauksia sivuston dokumenttioliomalliin (DOM) JavaScriptin funktioilla. (Mozilla ja yksittäiset avustajat 2020a.)

JavaScriptin käyttämä standardi on ECMAScript. ECMAScript nimi tulee standardia ylläpitävän ECMA International -yhdistyksen (the European Computer Manufacturer's Association) nimestä. Yhdistyksen sisäinen osasto, Technical Committee number 39 (TC39), on vastuussa uusien ECMAScript versioiden julkaisusta ja kehityksestä. Yhdistys pyrkii julkaisemaan vuosittain uusia versioita JavaScriptistä. (Mohan ja Prusty 2018, 1.)

### 2.1 TypeScript

TypeScript on avoimen lähdekoodin ohjelmointikieli, joka kääntyy JavaScriptiksi. TypeScript on niin kutsuttu "superset" JavaScriptistä eli se tarjoaa samat ominaisuudet kuin JavaScript sekä lisää siihen tyyppityksen. TypeScript-tiedostot päättyvät ".ts" tai ".tsx" -tiedostopäätteeseen riippuen siitä, että käytetäänkö ohjelmassa React-ohjelmistokehystä. (Vanderkam 2019, 1.)

JavaScript-koodi on validia TypeScriptiä. Niin kauan kuin JavaScript-ohjelmistossa ei ole syntaksivirheitä, niin se on myös silloin TypeScript-ohjelmisto. Tämä ominaisuus helpottaa olemassa olevien JavaScript-ohjelmistojen siirtämistä TypeScriptin syntaksiin. Eli JavaScript-ohjelmistot ovat myös TypeScript-ohjelmistoja, mutta asia ei ole sama toisinpäin. Tämä johtuu siitä, että TypeScript lisää syntaksia tyyppien määrittelyä varten. (Vanderkam 2019, 1.)

TypeScriptin suurin etu on vahva tyyppitys. Tyyppien määrittelemisen helpottaa kehitystyötä koska mahdolliset tyyppivirheet, jotka voisivat aiheuttaa bugeja, havaitaan useimmiten jo kirjoitusvaiheessa. TypeScriptin tyyppitarkistaja käy läpi muuttujat tallennuksen yhteydessä. Jos tyyppitarkastaja havaitsee ongelman, konsoliin tulee ilmoitus virheestä ja applikaatio saattaa kaatua käynnistäessä. Jos tyyppitarkastaja ei havaitse ongelmia, niin TypeScript käännetään JavaScriptiksi.

Kuva 1 näyttää esimerkin TypeScript-koodista. Muuttujan "greeting" tyyppi määritellään merkkijono. Kyseistä muuttujaa käytetään funktion "printText" parametrinä, joka on myös määritelty merkkijonoksi. Jos esimerkiksi "greeting"-muuttuja olisi määritelty numeroksi (number), TypeScript tyyppitarkastaja ilmoitaisi virheestä.

```
1  const greeting: string = 'Hello world'
2
3  function printText(text: string): void {
4    console.log(text)
5  }
6
7  printText(greeting)
```

KUVA 1. TypeScript-koodia. Tyypit määritellään TypeScriptissä kaksoispisteen jälkeen.

## 2.2 TypeScript ja React

TypeScriptiä voi käyttää Reactin kanssa, sillä React-ohjelmistokehys käyttää JavaScriptiä. Käyttöliittymät, jotka teimme Visma AI Worksille, käyttivät TypeScriptiä. Sovellukset luotiin create-react-app-työkalulla. Kyseinen komento luo valmiiksi applikaation pohjan. Työkalun parametreihin annetaan komento, jolla määritellään käyttääkö applikaatio TypeScriptiä vai ei. Kun komento on annettu, create-react-app-työkalu luo valmiin sovelluspohjan, jota voi lähteä muokkaamaan vapaasti.

## 3 REACT

React on Facebookin kehittämä avoimen lähdekoodin käyttöliittymäkomponenttikirjasto. Käyttöliittymäkomponentit luodaan Reactin avulla käyttäen JavaScriptiä. (React 2020, 1.) Reactin periaate on se, että luodaan uudelleenkäytettäviä käyttöliittymäkomponentteja. (React 2020, 2.) Reactin käyttöliittymäkomponentit hallinnoivat omaa tilaansa, jolloin niihin voidaan lisätä paljon toiminnallisuutta. Reactin käyttöliittymäkomponentit luodaan Reactin omalla JSX-syntaksilla, joka yhdistää JavaScriptin ja HTML:n. (React 2020, 3.) Reactia ylläpitää pääosin Facebook ja joukko yksittäisiä käyttäjiä sekä yrityksiä. (Dawson 2014.)

### 3.1 JSX

JSX on JavaScript-lisäosa, jossa on XML:n kaltainen syntaksi. Se yhdistää HTML:n ja JavaScriptin, jonka avulla Reactissa voidaan luoda elementtejä helposti. JSX helpottaa koodinlukua ja koodin muokkausta. JSX:ään voidaan lisätä muuttujia. Kun muuttujien tieto päivittyy, päivittyy myös käyttöliittymäkomponentti. JSX käännetään EcmaScriptiksi Babelin avulla. (Mardan 2017, 3.) Kuvassa 2 näytetään JSX-syntaksia.

```
const element = <h1>Hello, world!</h1>;
```

KUVA 2. JSX-syntaksia. Muuttujaan "element" on tallennettu JSX-syntaksia, joka yhdistää JavaScriptin ja HTML:n.

### 3.2 Komponentit

React on tehty komponentti-nimisistä kokonaisuuksista ja nämä kokonaisuudet esitetään DOMissa käyttäen Reactin DOM-kirjastoa. Komponentteja on mahdollista tehdä kahdella eri tavalla: luokkapohjaisesti tai funktionaalisesti. Tulemme käyttämään funktionaalista ohjelmointia tulevissa esimerkeissä. Kuvassa 3 on esimerkki funktionaalisesta komponentista Reactissa.

```
function Hello(props) {  
  return (  
    <div>Hello, {props.name}</div>  
  )  
}
```

KUVA 3. Esimerkki funktionaalista komponentista.

React näyttää komponenttinsa virtuaali-DOMissa. Tämä mahdollistaa lähes reaaliaikaisen näkymän päivityksen. Eli kun ohjelmoija ohjelmoi ja tallentaa muutokset, päivittyvät nämä muutokset heti selaimessa automaattisesti, eikä ohjelmoijan tarvitse kääntää ohjelmaa joka kerta uudestaan. (React 2020, 2.).

### 3.3 React Hooks

React julkaisi 6 helmikuuta, 2019 uuden version v16.8, joka toi mukanaan Hooks-ominaisuuden (Abramov 2019). Tämä päivitys mahdollisti funktionaalisen ohjelmoinnin Reactissa, joka on nopeampaa ja yksinkertaisempaa ohjelmoida.

React Hooks on kokoelma erilaisia funktioita, jotka korvaavat luokkapohjaisessa käytössä olleet elämänkaarifunktiot. React Hooks tarjoaa muutaman valmiin hookin heti käyttöön, kuten useState ja useEffect, mutta käyttäjä voi itse tehdä myös hookeja jos hän haluaa.

Kaksi käytetyintä hookiea ovat edellä mainitut useState ja useEffect. useStatea voi käyttää DOM-elementin sisällä. useState funktioon on sidottu muuttaja ja mahdollinen funktio, jota käytetään muuttujan tilaan muuttamiseen. (React 2020, 4.) Kuvassa 4 on esimerkki yksinkertaisesta React-komponentista, jossa käytetään useStatea lisäämään numeroon aina yksi lisää.

```
function countUp() {  
  const [count, setCount] = useState<number>(0)  
  
  return (  
    <>  
      <div>Count: {count}</div>  
      <button onClick={() => setCount(count + 1)}>+</button>  
    </>  
  )  
}
```

KUVA 4. Esimerkki count-laskurin päivittämisestä.

useEffectia käytetään sivuefektien luomiseen. (React 2020, 5.) Esimerkiksi kun komponentti aukeaa ensimmäistä kertaa, tulee ohjelman hakea ohjelmointirajapinnasta tietoa. Tähän voi käyttää useEffect-funktiota. Kuvassa 5 näytetään, miten tulostetaan selaimen konsoliin 'Hello' aina kun count-arvo päivittyy.

```
function countUp() {  
  const [count, setCount] = useState<number>(0)  
  
  useEffect(() => {  
    console.log('Hello')  
  }, [count])  
  
  return (  
    <>  
      <div>Count: {count}</div>  
      <button onClick={() => setCount(count + 1)}>+</button>  
    </>  
  )  
}
```

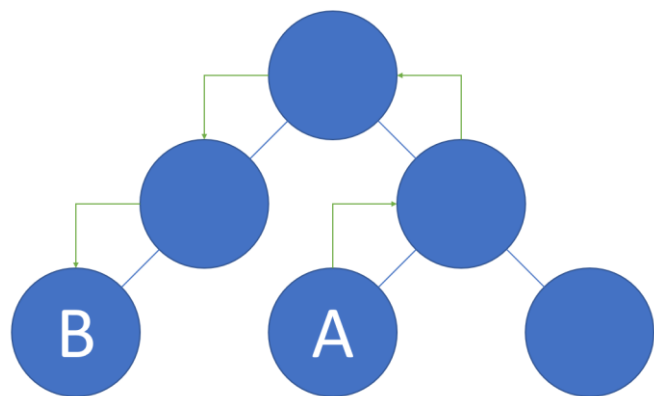
KUVA 5. Esimerkki Hello-tulostuksesta.

### 3.4 Redux

Redux on tilanhallintakirjasto, jolla hallitaan annettuja arvoja ohjelman sisällä. (Redux 2020.) Yleensä Reduxia ei kannata käyttää pienessä projektissa, koska tämä voi hankaloittaa tilanhallintaa, mutta laajemmissa projekteissa Redux on lähes pakollinen. Reduxin avulla pystyy helposti hallitsemaan laajan projektin arvoja monessa eri näkymässä.

Kuviossa 1 näytetään kuvio ohjelmistosta ilman Reduxia. Ohjelman viedessä tietoa kohdasta A pisteeseen B, joutuu tämä kiertämään monen mutkan kautta viedäkseen tiedon.

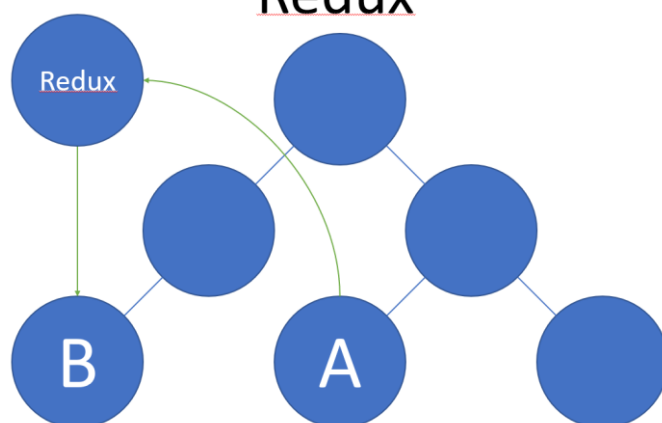
## Ilman Reduxia



KUVIO 1. Ohjelma ilman Reduxia. Tieto täytyy kuljettaa monen komponentin lävitse.

Kuviossa 2 tiedonvienti kohdasta A kohtaan B helpottuu paljon Reduxin takia, ja ohjelmasta ei tarvitse tehdä yhtä monimutkikasta kuin kuviossa 1.

## Redux



KUVIO 2. Ohjelma Reduxilla. Redux tarjoaa yhtenäistetyn tilan, jolloin tietoa ei tarvitse kuljettaa komponenttien läpi.

## 4 TARVITTAVAT OMINAISUUDET JA SUUNNITTELU

Asiakkaan vaatimuksena oli toteuttaa käyttöliittymät, joiden avulla he pystyvät hallinnoimaan myyntikampanjoita. Tavoitteena oli toteuttaa kaksi käyttöliittymää: Management-käyttöliittymä ja Agent-käyttöliittymä. Management-käyttöliittymästä, tuli pystyä hallitsemaan myyntikampanjoita sekä niiden soittolistaa. Management-käyttöliittymän vaatimuksena oli myös, että se sisältäisi työajanseurantatyökalun, sekä raporttinäkömään, josta näkisi raportteja myyntikampanjoista. Toteutuksen täytyi olla verkkoapplikaatio.

Toinen käyttöliittymistä oli myyntiagenteille suunnattu Agent-käyttöliittymä, jossa myyntiagentti näkisi asiakkaan tiedot ja kampanjan tiedot. Agent-käyttöliittymässä myyntiagentti voisi merkitä kontaktille tilauksia, sekä kysellä myyntikampanjalle määriteltyjä kysymyksiä. Käyttöliittymässä tulisi olla mahdollista seurata sekä lisätä työvuoroja yksittäiselle myyntiagentille. Toteutuksen täytyi olla verkkoapplikaatio.

### 4.1 Management-käyttöliittymä

Management-käyttöliittymä haluttiin toteuttaa verkkoapplikaationa, mistä työjohto voisi helposti millä tahansa laitteella hallinnoida myyntikampanjoita. Management käyttöliittymä tarvitsi kolme pääominaisuutta: kampanjanhallinta, raportointi ja työajanseuranta.

Kampanjahallintänäkömystä pitää pystyä luomaan kampanjoita sekä selata myyntikampanjoita helposti listalta. Yksittäisen kampanjan hallintaan tarvitaan monta erilaista ominaisuutta: kampanjan tietojen-, kontaktien-, tuotteiden-, ja kysymyshallinta. Kontakteja pitää pystyä lisäämään manuaalisesti tai niitä voidaan tuoda CSV-tiedostosta.

Raportointinäkömään ominaisuuksia ovat erilaisten raportointipohjien perusteella tehtävät raportit, jossa voi valita kampanjan, toimiston ja aikavälin perusteella. Työajanseurantatyökalulla voi seurata työntekijöiden työaikoja sekä puhelutöiden aikoja.

Näiden vaatimusten perusteella valitsimme React-kirjaston, jonka ympärille lähdimme suunnittelemaan sovelluksen arkkitehtuuria. Huomasimme aikaisin projektin alussa, että sovellusta olisi vaikea saada sopimaan mobiilikäyttöliittymään, joten asiakkaan kanssa totesimme, että sovellusta ei suunnitella ollenkaan mobiilille. Jotta sovellus olisi moderni ja hyvännäköinen, käytimme sovelluksessa Material UI -komponenttikirjastoa, joka mahdollisti sovelluksen nopean ja helpon kehityksen.

## 4.2 Agent-käyttöliittymä

Agent-käyttöliittymän vaatimuksena oli tehdä toimiva ja nykyaikainen käyttöliittymä, joka olisi räätälöity asiakkaiden vaatimusten mukaan. Käyttöliittymän tuli olla verkkoapplikaatio, joka näytettäisiin ICC-managerin sisällä olevassa Chromium-ikkunassa. Käyttöliittymässä tulisi olla kaksi eri sivua, joista löytyy myyntitilanteessa tarvittavat tiedot myyntikampanjasta sekä asiakkaasta. Ensimmäiseltä sivulta tuli löytyä myyntikampanjakohtainen myyntipuhe, kontaktin tiedot, lopputulokset sekä extrakentät. Toisella sivulla tulisi olla myyntikampanjakohtaiset tuotteet sekä kysymykset. Agent-käyttöliittymän tulisi olla responsiivinen. Agent-käyttöliittymän tiedot haettaisiin rajapinnan kautta. Lopuksi päivitettyt tiedot lähetettäisiin rajapinnalle.

Ensimmäinen sivu, joka nimettiin kehitysvaiheessa kontakti-sivuksi, sisälsi paljon erilaisia ominaisuuksia. Kontakti-sivulla olevat kontaktin tiedot tuli olla muokattavissa. Lopputulokset tuli myös näkyä painikkeina lopputuloslistan alapuolella, jos ne oli määritelty Management-näkymässä pikanäppäimiksi. Extrakentät määriteltiin näkyviksi Management-käyttöliittymän kautta. Lopputuloksen kirjaamisen suhteen tuli tehdä tarkistuksia, riippuen siitä minkä lopputuloksen käyttäjä oli valinnut. Jos lopputuloksen tarkistus ei mene läpi, niin käyttäjälle tuli näyttää varoitusviesti.

Toinen sivu, joka nimettiin Tuotteet ja kysymykset -sivuksi, sisälsi paljon monimutkaista toiminnallisuutta. Tällä sivulla tuli näkyä tuotteet sekä kysymykset. Sivulla näkyvät kysymykset tuli näyttää vain silloin, kun niille määritelty "ehto" tai "riippuvuus" täyttyy. Kysymykset voivat olla riippuvaisia tuotevalinnasta, kontaktin tiedoista tai muista kysymyksistä. Kysymyksillä ei ollut pakko olla riippuvuutta,

jolloin ne näkyisivät sivulla automaattisesti. Riippuvuudet määritetään Management-näkymässä.

## 5 MANAGEMENT-NÄKYMÄN TOTEUTUS

Management-näkymässä on sovelluksen päänäkyvä, jonka ympärillä koko projekti pyörii. Tässä sovelluksessa luodaan ja hallitaan kampanjoita, tuotetaan näistä kampanjoista raportteja sekä seurataan agenttien työvuoroja ja työnto-  
teutumista.

Sovellus on arkkitehtuurisesti helposti laajennettava ja muokattava. Kuviossa 3 esitetään sovelluksen arkkitehtuuria. Rakenne kuvaa miten uusia osia voi lisätä tuotteeseen tai kuinka ne voi ottaa pois vaikuttamatta muihin osiin.



KUVIO 3. Yksinkertainen kuvaus sovelluksen arkkitehtuurista

### 5.1 Kirjautuminen

Ennen kuin sovellusta voi käyttää, tulee käyttäjän kirjautua sovellukseen. Kirjautumisessa käytetään Visma AI Worksin kehittämää kirjautumisvarmenninta, jota käytetään monessa Visma AI Worksin kehittämässä tuotteessa. Käyttäjä kirjautuu ohjelmaan sisään painamalla kirjautumisnappulaa. Tämä ohjaa käyttäjän kirjautumissivulle, jossa käyttäjä antaa kirjautumistietonsa. Kirjautumisen onnistuttua palvelu ohjaa käyttäjän takaisin management-näkymään, jossa kirjautumisvarmentimesta saatu token varmistetaan ja tallennetaan jatkokäyttöä varten.

Uloskirjautumisessa käyttäjä painaa samaa nappulaa, mutta tämä vain poistaa tokenin muistista, merkkää uloskirjautumisen ja ohjaa uudelleen käyttäjän kotisivulle.

## 5.2 Navigaatio

Käytämme sovelluksessa React Routeria, jonka avulla siirrämme käyttäjän näkymästä toiseen.

Päänäkymät ovat

- Koti
- Kampanjat
- Raportit
- Kalenteri.

Kampanjat-näkymässä käytämme myös alinavigointina kampanjalistaa, joka uudelleenohjaa käyttäjän haluttuun yksittäiseen kampanjanäkymään. Tämän kampanjanäkymän sisällä on myös välilehtinappeja, jotka muuttavat ohjelman sisällä olevaa näytettävää välilehteä.

## 5.3 Kampanjat-näkymä

Kampanjat-näkymässä on muutamia erilaisia komponentteja, jotka ovat lista kampanjoista, hakuominaisuus, jossa voi kampanjan nimellä suodattaa listaa sekä kampanjan luontivalikko.

Kampanjanluonnissa käyttäjän pitää antaa kampanjalle nimi ja lyhenne sekä mahdollinen kopiopohja, josta kampanja voidaan kopioida. Tämä kopiopohja luodaan kampanjasta, joka on jo olemassa. Kopioidessa kampanjapohjaa saa uusi kampanja kaikki pohjakentät paitsi nimen ja lyhenteen. Tämä ominaisuus on hyvin tärkeä kausimarkkinoinnissa. Hyvä esimerkki tästä on neljä kertaa vuodessa myytävä lehti, jonka jokaisessa julkaisussa muuttuu vain muutama asia. Tämä nopeuttaa ja helpottaa työntekoa huomattavasti.

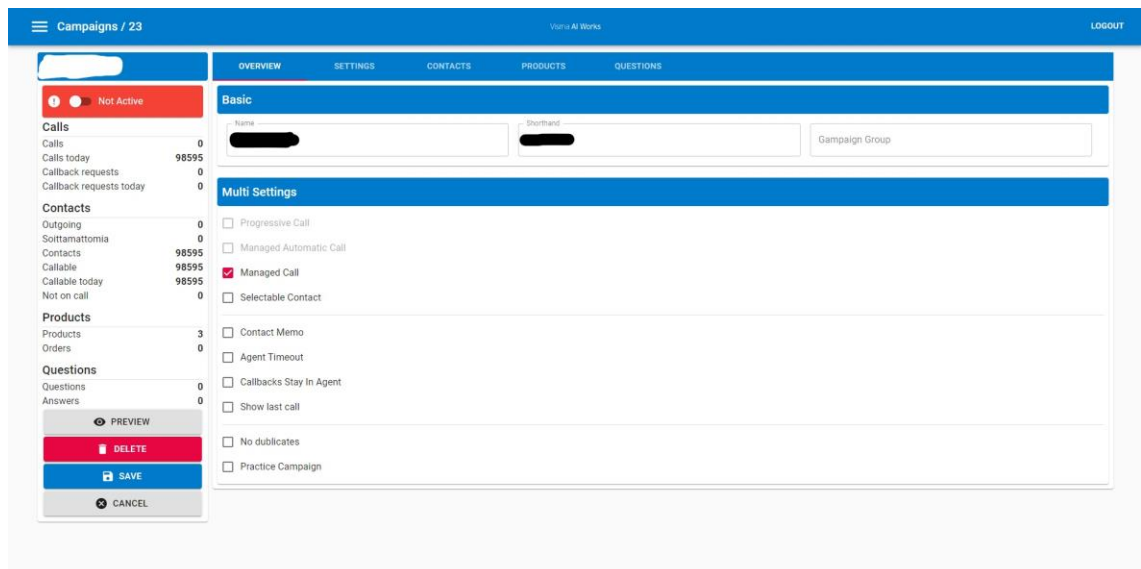
Pääkomponentissa on kampanjalista. Tässä komponentissa näytetään seuraavat asiat: kampanjan aktivointi- ja pysäytyspainike, kampanjan yksilöivä tunnus, nimi, lyhenne ja ryhmä. Lisäksi kampanjan tilastoista esitetään seuraavat numeraaliset kohdat: soitettavat kohteet, soitetut kohteet, kysymykset, tuotteet ja vastaukset. Näiden tietojen avulla soittokampanjoiden seuranta on helppoa ja nopeaa. Kuvassa 6 on esimerkki Kampanjalistasta, joka on täytetty erilaisilla kampanjoilla.

Actions	id	Name	Shorthand	Group	Contacts	Calls	CT	Q	P	A	Date
II	23	[REDACTED]	[REDACTED]		98595	0	0	0	3	0	2020-07-30T12:11:04.343
II	12	[REDACTED]	[REDACTED]		5843	15	0	0	1	0	2020-07-15T14:42:36.017
II	28	[REDACTED]	[REDACTED]		1	0	0	2	1	0	2020-09-18T13:01:20.85
II	29	[REDACTED]	[REDACTED]		259	0	0	0	2	0	2020-09-18T13:03:40.72
II	34	[REDACTED]	[REDACTED]		1	0	0	0	0	0	2020-09-18T13:20:12.553
II	14	[REDACTED]	[REDACTED]		712	0	0	1	0	0	2020-07-15T11:55:50.83
II	15	[REDACTED]	[REDACTED]		7643	0	0	0	0	0	2020-07-15T11:59:14.843
II	13	[REDACTED]	[REDACTED]		100000	0	0	6	0	0	2020-07-15T11:46:56.363
II	17	[REDACTED]	[REDACTED]		72	62	0	6	2	0	2020-07-20T07:45:15.457
II	5	[REDACTED]	[REDACTED]		3	4	0	2	2	0	2018-04-19T14:40:35.07

KUVA 6. Kampanjalista.

#### 5.4 Yksittäinen kampanjanäkymä

Kampanjanäkymässä on kaksi pääkomponenttia: hallintapalkki sekä pääinfopaneeli. Vasemmalla sijaitsee hallintapalkki, jossa käyttäjä näkee kampanjan statistiikkaa tarkemmin, sekä neljä nappulaa. Kuvassa 7 näytetään mahdollinen esimerkki kampanjasta.



KUVA 7. Yksittäinen kampanja.

Preview-nappulalla saa Agenti-näkymän esikatselun esille. Tämän avulla kampanjan tekijä voi nähdä ennakkoon, mitä agentti näkee, kun agentti saa kampanjan työnalle. Tämä näkymä on suora kopio agenttinäkymästä, josta kerrotaan myöhemmin lisää. Delete-nappula poistaa kampanjan, save-nappula tallentaa kampanjan, sekä cancel-nappula peruttaa näkymästä pois.

Pääinfopaneelin sisältö muuttuu aina valitun välilehden mukaan. Ensimmäisenä on Overview-välilehti, joka näytetään ensimmäisenä, ja johon on kopioitu muiden välilehtien tärkeimmät osat pikaista käyttöä varten. Muut välilehdet ovat Settings, Contacts, Products ja Questions.

#### 5.4.1 Kampanjan asetukset

Asetukset eli Settings-välilehti toimii ensimmäisenä välilehtenä kampanjanhallintanäkymässä. Tässä näkymässä on neljä omaa välilehteä: Basic Info, Call Settings, Completion Codes ja Extra Settings, joissa säädetään kampanjaan liittyviä asetuksia.

Basic Info -välilehdessä pystytään muokkaamaan kampanjan nimeä, lyhennettä ja ryhmää. Näitä käytetään kampanjan hallinnassa, jotta samanlaiset kampanjat voi helposti erottaa muista kampanjoista. Seuraavaksi on Sales Speech eli

myyntipuhe. Tähän laitetaan teksti, jonka agentti lukee ääneen tehdessään soittoa asiakkaalle. Viimeisenä muokattavana kohteena on Description eli muistiinpanot. Tähän kampanjan tekijä voi kirjoittaa asioita, joita agentin kannattaa muistaa asiakkaalle soittaessa. Kuvassa 8 on esimerkki Basic Info -välilehdestä, jossa kampanjan nimi, lyhenne ja Description on täytetty.

The screenshot displays the 'Basic Info' configuration page for a campaign. On the left sidebar, there are statistics for 'Calls' (98595 today), 'Contacts' (98595 total, 98595 callable), 'Products' (3 total), and 'Questions' (0 total). The main content area is titled 'Basic Info' and includes a progress indicator with four steps: 1. Basic Info (active), 2. Call Settings, 3. Completion Codes, and 4. Extra Settings. The 'Basic' section contains input fields for 'Name', 'Shortname', and a dropdown for 'Campaign Group'. Below this is the 'Sales Speech' section with a large text area. The 'Description' section also has a large text area. At the bottom, there are four buttons: 'PREVIEW', 'DELETE', 'SAVE', and 'CANCEL'.

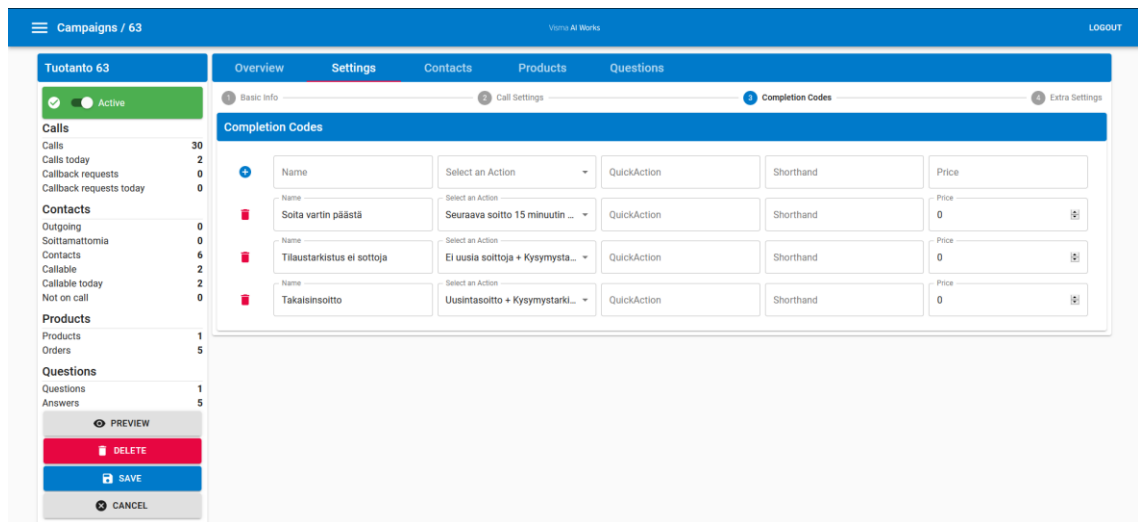
KUVA 8. Basic Info -välilehti.

Seuraava välilehti on Call Settings. Tässä näkymässä käyttäjä voi muuttaa soitto- ja lopetuspäivää sekä soittoaikoja. Esimerkiksi maanantaisin voi soittaa kello 9.00 - 11.00 sekä klo 12.00 – 15.00 välisenä aikana. Näitä valintoja voi tehdä jokaiselle viikonpäivälle. Muita soiton asetuksia ovat uusinta-soittojen määrä, mahdollinen hinta per soitto, jonka kampanjan tilaaja pitää maksaa, sekä soittajan palkkio. Kuvassa 9 esitetään miltä Call Settings -välilehti näyttää

KUVA 9. Call Settings -välilehti.

Tärkein kampanjan asetus on kuitenkin Multi Settings -kohta, jossa määritellään soittotyyppi. Nämä tyypit ovat robotin tekemä automaattisoitto, jossa vastannut asiakas siirretään agentille, hallinnoitu soitto, jossa yksi kerrallaan agentti soittaa asiakkaalle, sekä valintasoitto, jossa agentti voi itse päättää, mille kohteelle soittaa. Asetuksissa voi myös hallita laaduntarkastusparametreja. Nämä parametrit ovat sidottu tekoälyn tekemiin soittoihin ja muuttaa toimintaansa näiden perusteella. Seuraavasta osiosta löytyy Extra URL, tähän annetaan linkki, jonka agentti saa tehtyään puhelutyön. Tämä linkki voi olla esimerkiksi palaute kaavio, joka täytetään jokaisen puhelun jälkeen. Viimeinen osia tässä näkymässä on agenttilista. Tämä lista kertoo kuka voi ja kuka ei voi soittaa tässä kampanjassa. Vakiona kaikki agentit on laitettu tekemään kampanjaa.

Kolmas välilehti on Completion Codes eli valmismvastaukset. Tässä näkymässä määritellään, mitä pikavastauksia agentilla voi olla puhelun lopussa. Esimerkiksi "ei vastannut", tässä tilanteessa asiakas ei vastannut puhelimeen ja puhelu katkeaa. Agentti painaa pikavastausnappulaa ei vastannut ja voi siirtyä seuraavaan asiakkaaseen. Tällaisia erilaisia pikavastauksia on kymmeniä mahdollisia. Kuvassa 10 näytetään miltä Completion Codes -välilehti voi näyttää.



KUVA 10. Completion Codes -välilehti.

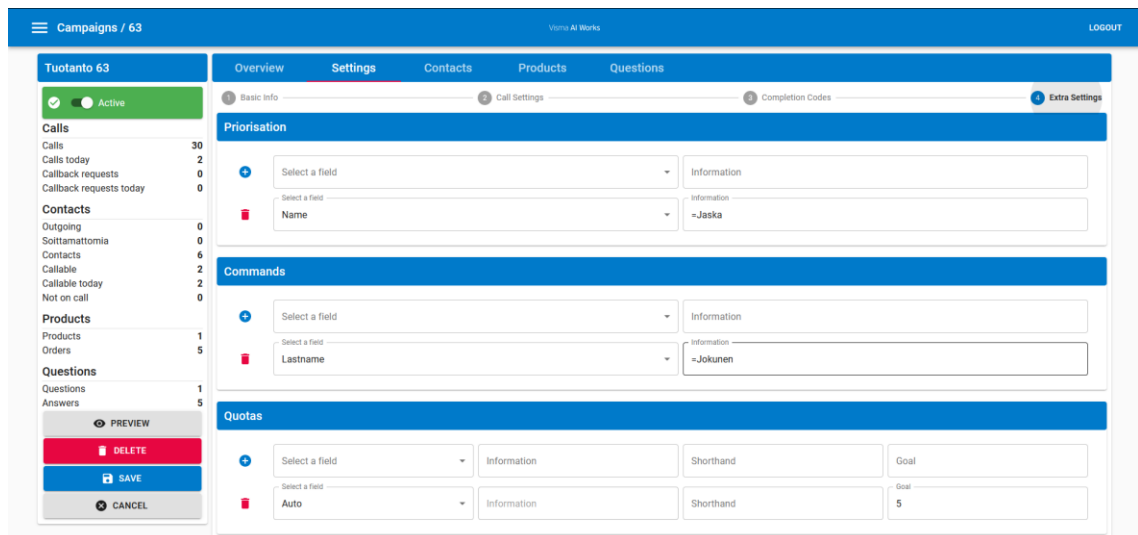
Viimeinen asetusten välilehti on Extra Settings. Tässä näkymässä on kolme erilaista asetusta, joilla voi hienosäätää kampanjaa tarkemmin.

Priorisation-kohdassa määritellään, mitä kontakteja lähdetään soittamaan ensin. Kuvasta 11 nähdään Priorisation-kohdasta, että henkilöitä, joiden nimi on Jaska, lähdetään soittamaan ensimmäisenä läpi.

Commands on muuten sama kuin Priorisation, sillä poikkeuksella että vain kriteerit täyttäneet kontaktit soitetaan läpi. Esimerkiksi kuvassa 11 kontaktin sukunimi pitää olla Jokunen, että hänelle soitetaan.

Viimeisenä on Quotas. Tällä määritellään, milloin kampanja pysäytetään. Kohteessa täytyy valita, mitä asiaa seurataan ja mikä on mahdollinen kriteeri, joka pysäyttää kampanjan. Kuvassa 11 nähdään, että jos viisi autoa on myyty, kampanja pysähtyy.

Kuvassa 11 on valmiiksi määritelty kampanja. Tässä kontaktit, joiden sukunimi on Jokunen, haravoidaan läpi. Näistä kontakteista, ne joiden nimi on Jaska, haravoidaan ensimmäisenä läpi. Kun puheluita tehdään, seurataan autojen myyntiä ja jos autoja on myyty viisi kappaletta, pysäytetään kampanja.



KUVA 11. Extra Settings -välilehti

## 5.4.2 Kontaktit

Contacts-välilehdestä löytyy useita komponentteja, jotka pystytään jakamaan kolmeen eri osaan: laajentuvat toiminnot, kontaktilista ja soittolista. Laajentuvat toiminnot -osiossa on neljä painiketta: uuden kontaktin luonti, extrakenttien määrittely, massakontaktien tuonti ja kontaktien haku.

Uuden kontaktin luonnissa määritellään kontaktin tiedot, esimerkiksi nimi, sukunimi, puhelinnumero ja osoite. Kontaktilla on useita eri tietoja, ja yksi näistä on extrakenttä. Kentät ovat kampanjakohtaisesti sidottuja tietoja kontakteista.

Kontakteja ei tarvitse kuitenkaan yksitellen tuoda ohjelmaan vaan tähän voi käyttää kontaktin massatuontia, jossa CSV-tiedostossa olevat kontaktit tuodaan kampanjaan.

Kontaktilista-kohdasta nähdään, mitä kohteita on ladattu tähän kampanjaan. Kohteita pystytään muokkaamaan ja poistamaan. Tehdyistä soitoista on myös lista nimeltä Calls, näitä elementtejä ei voi muokkaa tai poistaa. Kuvassa 12 näytetään Contacts-välilehden ulkoasua.

The screenshot shows the 'Contacts' tab in the Voima AI Works system. The sidebar on the left indicates 30 calls, 2 calls today, and 0 callback requests. The main content area displays a table of contacts with the following data:

Actions	Name	Lastname	E-Mail	Phonenumber	Phonenumber 2	Delivery Address	Postal Number	Postal Area	Company	WWW
[Red] [Blue]	Jani			[Redacted]						
[Red] [Blue]	Mika			[Redacted]						
[Red] [Blue]	Soita vartin päästä			[Redacted]						
[Red] [Blue]	Tees			[Redacted]						
[Red] [Blue]	Testti	Nimi		[Redacted]						
[Red] [Blue]	Tilaus	Soita takaisin		[Redacted]						

Below the contacts table, there is a 'Calls' table with the following data:

Agent	Contact	Call time	Call start	Call end	UCID
	[Redacted]	0	0001-01-01T00:00:00	0001-01-01T00:00:00	
	[Redacted]	0	0001-01-01T00:00:00	0001-01-01T00:00:00	
	[Redacted]	0	0001-01-01T00:00:00	0001-01-01T00:00:00	
	[Redacted]	0	0001-01-01T00:00:00	0001-01-01T00:00:00	

KUVA 12. Contacts-välilehti.

### 5.4.3 Tuotteet

Neljäs välilehti on Products. Tällä välilehdellä on tuotteen Lisäys-osio, tuotelista, sekä tilauslista. Tuotteen Lisäys-osiossa voi tuotteelle määrittellä tuotekohtaisia tietoja, kuten hinta ja nimi. Tuotteet ovat kampanjakohtaisia, ja jokaisella kampanjalla on omat tuotteensa. Välilehdeltä löytyy myös Tilaukset-osia eli Orders. Tässä osiossa löytyy lista tuotteista, joita on myyty, ja ketkä niitä ovat ostaneet. Kuvassa 13 näytetään Products-välilehden ulkoasua.

The screenshot shows the 'Products' tab in the Voima AI Works system. The sidebar on the left indicates 30 calls, 2 calls today, and 0 callback requests. The main content area displays a table of products with the following data:

Action	Name	Group	Price	Product Code	CampaignCode	Extra 1	Extra 2	Extra 3
[Blue] [Red]	Auto	ajoneuvo	8000	koodi1				

Below the products table, there is an 'Orders' table with the following data:

First name	Lastname	Address	Postal Area	Postal Code
Jani				
Mika				
Soita vartin päästä				
Tees				
Tilaus	Soita takaisin			

KUVA 13. Products-välilehti

#### 5.4.4 Kysymykset

Kysymykset-välilehdellä käyttäjä voi luoda kysymyksiä kyseiselle myyntikampanjalle. Kysymykset näkyvät Agent-käyttöliittymässä. Kysymyksille voidaan lisätä riippuvuuksia. Kysymyksistä sekä kysymysten riippuvuuksista kerrotaan enemmän kappaleessa 6.3. Tämä on viimeinen osio yksittäisen kampanjan näkyvässä.

#### 5.5 Raporttinäkymä

Raporttien perusteella asiakas pystyy kerätä helposti tietoja, miten kampanjat ovat toimineet ja näiden perusteella laskuttaa heidän asiakkaitansa. Raportteja käytetään myös sisäisesti seuratakseen miten agentit kehittyvät ja missä kategorioissa he voivat kehittyä. Raporttipohjia on kehitetty 5 kappaletta:

- Live, tätä käytetään tämänhetkisten tilanteiden nopeaan seurantaan, jota yleensä työjohto käyttää nopeaan tilanteen seurantaan.
- Productivity, tätä käytetään, kun halutaan seurata agenttien tuottavuutta.
- Orders, kun asiakkaan asiakas haluaa tietää tilausten tilastoja, käytetään tätä. Tilastoissa tuotetta Z myytiin 10, joista agentti saa 5% provisiota. Tätä provisiota maksetaan yhteensä 3678€ yhteensä ja niin edelleen.
- Contacts, Hakee kontaktien tilanteet per kampanja ja agentti. Esimerkiksi kampanjassa X on yhteensä tehty 100 soittoa, joista agentti y on tehnyt 10. Tämän agentin keskimääräinen puhelukesto on 10 minuuttia ja 35 sekuntia.
- Registers, tässä raportissa esitetään tietoja, miten puhelussa on käynyt, esimerkiksi kuinka moni ei vastannut, vastasi, mutta ei tilannut tuotetta ja vastasi ja teki tilauksen. Puheluiden lopputilanteet ovat dynaamisia, joten näitä kaikkia ei pysty listata.

Kun jokin näistä pohjista on valittu voi käyttäjä valita halutessaan mistä kampanjasta, mikä toimisto ja miltä aikaväliltä tämä raportti luodaan. Kun raportti on luotu voi tämän myös tulostaa ulos CSV-tiedostona ulos. CSV-tiedoston on tarkoituksena antaa asiakkaan asiakkaalle, kun tämä on siistitty turhista tiedoista pois. Kuva 14 on esimerkki raporttinäkymästä, jossa Orders-pohja on valittu.

The screenshot shows a web application interface for reports. On the left is a sidebar with filters: 'Select a report' (Orders), 'Office', 'User', 'Campaign', and 'Listing' (All, Not Canceled, Canceled, Expanded, Timeframe). At the bottom of the sidebar are 'CREATE REPORT' and 'CREATE CSV' buttons. The main area displays a report titled 'Compined - 28/9/2020 - 28/9/2020' with a table of product and campaign data. Below it are two test reports, 'Testi 1' and 'Testi 2', each with a table of agent and product data.

Product Code	Campaign Code	Extra Code	Margin %	Product	Extra 1	Extra 2	Extra 3	Campaign Name	Order Cost
0	0	0	10	14				Testi 1	129
0	0	0	22	64				Testi 2	100
									78
									229

Product Code	Campaign Code	Extra Code	Margin %	Product	Extra 1	Extra 2	Extra 3	A-ID	Agent Name	Created	Ci
0	0	0	45	Kahvi				0	Agentti 1	Wed Oct 28 2020 12:05:49 GMT+0200 (Itä-Euroopan normaaliaika)	W
0	0	0	123	Matto				1	Agentti 2	Wed Oct 28 2020 12:05:49 GMT+0200 (Itä-Euroopan normaaliaika)	W
											2

Product Code	Campaign Code	Extra Code	Margin %	Product	Extra 1	Extra 2	Extra 3	A-ID	Agent Name	Created	Ci
0	0	0	45	Kahvi				0	Agentti 1	Wed Oct 28 2020 12:05:49 GMT+0200 (Itä-Euroopan normaaliaika)	W
0	0	0	123	Matto				1	Agentti 2	Wed Oct 28 2020 12:05:49 GMT+0200 (Itä-Euroopan normaaliaika)	W
											2

KUVA 14. Raportointinäkömä

## 5.6 Kalenteri-näkymä

Kalenteri-näkymässä käyttäjä voi tarkastella kaikkien myyntiagenttien työvuoroja, toteutuneita työaikoja, sovittujen uusintasoittojen ajankohtia sekä sisäänkirjautumisia. Management-käyttöliittymän kalenteri-sivu on hyvin samankaltainen kuin Agent-näkymässä oleva kalenteri-sivu. Management-käyttöliittymässä oleva kalenteri-sivu on tarkoitettu pääkäyttäjälle, joka voi hallinnoida ja muokata kalenterin tietoja. Suurin ero Management- ja Agent-käyttöliittymien kalenteri-sivuilla on se, että Management-käyttöliittymässä voi muokata, poistaa ja lisätä työvuoroja kaikille agenteille, mutta Agent-käyttöliittymässä myyntiagentti voi ainoastaan tarkastella omia tietojaan, sekä ainoastaan lisätä työvuoroja itselleen, mutta ei poistaa niitä.

## 6 AGENT-NÄKYMÄN TOTEUTUS

Agent-näkymä on myyntiagenteille tarkoitettu käyttöliittymä, joka tulee puhelinoitoon yhteydessä näkyviin myyntiagentille. Agent-näkymästä käyttäjä voi tarkastella asiakkaan tietoja, myyntikampanjan kysymyksiä sekä myyntikampanjaan kuuluvia tuotteita. Käyttöliittymä on toteutettu ReactJS -ohjelmistokehyksellä ja se käyttää Redux -tilanhallintakirjastoa. Käyttöliittymä on sidoksissa Management hallintapaneeliin, josta voidaan hallita tietoja, jotka näkyvät Agent-käyttöliittymässä. Agent-näkymään pääsee ainoastaan Visma AI Worksin kehittämän ICC Manager -asiakaspalvelujärjestelmän sisällä olevassa Chromium-ikkunassa. Agent-näkymä on asennettu palvelimelle, johon ICC Manager ottaa yhteyttä. Kun yhteys on saatu ja tunnistautuminen on onnistunut, Agent-näkymä avautuu ICC Managerin Chromium-ikkunaan.

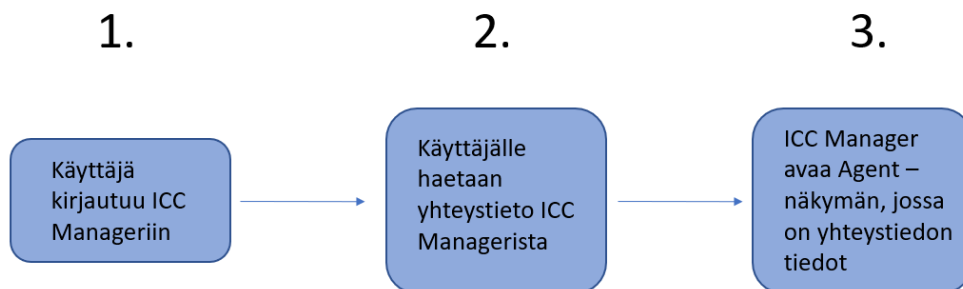
Agent-näkymässä on kaksi sivua, joista löytyy tietoa liittyen kontaktiin ja myyntikampanjaan. Ensimmäinen sivu on Kontakti-sivu. Siinä käyttäjä voi tarkastella kontaktin yhteystietoja, ja kirjata myyntitapahtuman lopputuloksia. Toinen sivu on Tuotteet ja kysymykset-sivu, jonka nimestä voi päätellä, että sieltä löytyy myyntikampanjaan kuuluvat tuotteet sekä kampanjaan kuuluvat kysymykset. Kuva 15 näyttää, miltä Agent-käyttöliittymän Kontakti-sivu näyttää.

KUVA 15. Agent-käyttöliittymän Kontakti-sivu.

## 6.1 Kirjautuminen

Agentti-käyttöliittymään kirjaudutaan Visma AI Worksin kehittämän ICC Manager -asiakaspalvelujärjestelmän kautta. Ensiksi myyntiagentin on kirjauduttava ICC Manageriin, jonka jälkeen hänelle tarjotaan yhteystietoa, jolle soitetaan. Kun ICC Manager on tarjonnut yhteystiedon, myyntiagentille avautuu Agent-näkymä.

Agent-näkymä käyttää Visma AI Worksin omaa tunnistautumis-palvelua. Tunnistautuminen tapahtuu siten, että ICC Manager tarjoaa ennalta haettua tokenia, ja token vahvistetaan Agent-näkymässä. Kun token on vahvistettu voimassaolevaksi ja validiksi, niin Agent-näkymään avautuu tiedot asiakkaasta, sekä kampanjan kysymykset ja tuotteet. Jos token ei ole validi, näytetään 401 Forbidden -virheviesti käyttäjälle. Kuvio 4 näyttää kirjautumisprosessin vaiheittain.



KUVIO 4. Kuvio myyntiagentin kirjautumisesta käyttöliittymään

## 6.2 Kontakti-sivu

Kontakti-sivulla näkyvät asiakkaan yhteystiedot, mahdolliset lopputulokset myyntitapahtumalle, extrakentät sekä myyntipuhe. Kontakti-sivu on ensimmäinen sivu, joka avautuu käyttäjälle Agent-käyttöliittymään. Käyttäjä voi muokata asiakkaan yhteystietoja, jotka näkyvät Kontakti-sivulla. Tämä on hyödyllistä esimerkiksi silloin, jos asiakas on muuttanut, eikä osoite ole oikea nykyisissä tiedoissa.

Lopputulokset näkyvät listana sivun oikeassa reunassa. Lopputulokset ovat ennalta määritettyjä lopputuloksia myyntitapahtumalle. Kirjaaminen tapahtuu aina

myyntitapahtuman lopuksi. Koska ajankäyttö on hyvin tarkkaa puhelinmyynnissä, niin lopputuloksien kirjaamisen tulee olla yhden napinpainalluksen takana, jos niin halutaan. Siksi on mahdollista määritellä Management-näkymässä lopputulokseen myös pikanäppäin, jolloin lopputulos näkyy lopputuloslistassa, sekä pikanäppäimenä listan alapuolella. Näin käyttäjä voi nopeuttaa toimintaansa. Sen sijaan, että käyttäjä joutuisi valitsemaan pudotusvalikosta lopputuloksen ja painamaan tallennusnappia, hän voi painaa vain yhtä nappia.

Extrakenttä voi olla jotain lisätietoa asiakkaaseen liittyen. Se näkyy Kontakti-sivun alaosassa. Extrakenttien näkyvyys määritellään Management-näkymässä. Extrakenttiä voi olla maksimissaan kymmenen kappaletta näkyvissä.

### 6.3 Tuotteet ja kysymykset -sivu

Tuotteet ja kysymykset -sivu on toinen Agent-käyttöliittymän sivuista, jossa käyttäjä voi nähdä yhteystiedon tilaukset kyseiselle myyntikampanjalle, myyntikampanjaan kuuluvat kysymykset ja vastaukset sekä myytävät tuotteet. Sivun on jaettu kahteen puoliskoon: vasemmalla puolella näkyy myyntikampanjan tuotteet, oikealla puolella näkyvät kysymykset ja vastaukset. Kuva 16 näyttää miltä Tuotteet ja kysymykset -sivu näyttää.

The screenshot shows the 'Tuotteet ja kysymykset' page. The top navigation bar includes 'Provad Testi / 67', 'KONTAKTI', and 'TUOTTEET JA KYSYMYKSET'. The page is divided into two main sections: 'Tuotteet' on the left and 'Kysymykset' on the right.

**Tuotteet section:**

- Auto:** A form with fields for 'First name' (Teppo), 'Last name' (Testaaja), 'Postal area' (Tampere), 'Postal number' (33100), and 'Delivery address'. A '+' sign is below the form.
- Pitsa:** A form with fields for 'First name' (Matti), 'Last name' (Meikäläinen), 'Postal area' (Tampere), 'Postal number' (33100), and 'Delivery address'. A '+' sign is below the form.

**Kysymykset section:**

0. Ostiko auton? !
  - Kyllä
  - Ei
1. Tilasiko pitsan? !
  - Kyllä
  - Ei
2. Auton vuosimalli? !
  - 2010 tai uudempi
  - 2009 tai vanhempi
3. Monivalinta kysymys !
  - a
  - b
  - c

KUVA 16. Agent-käyttöliittymän Tuotteet ja kysymykset -sivu.

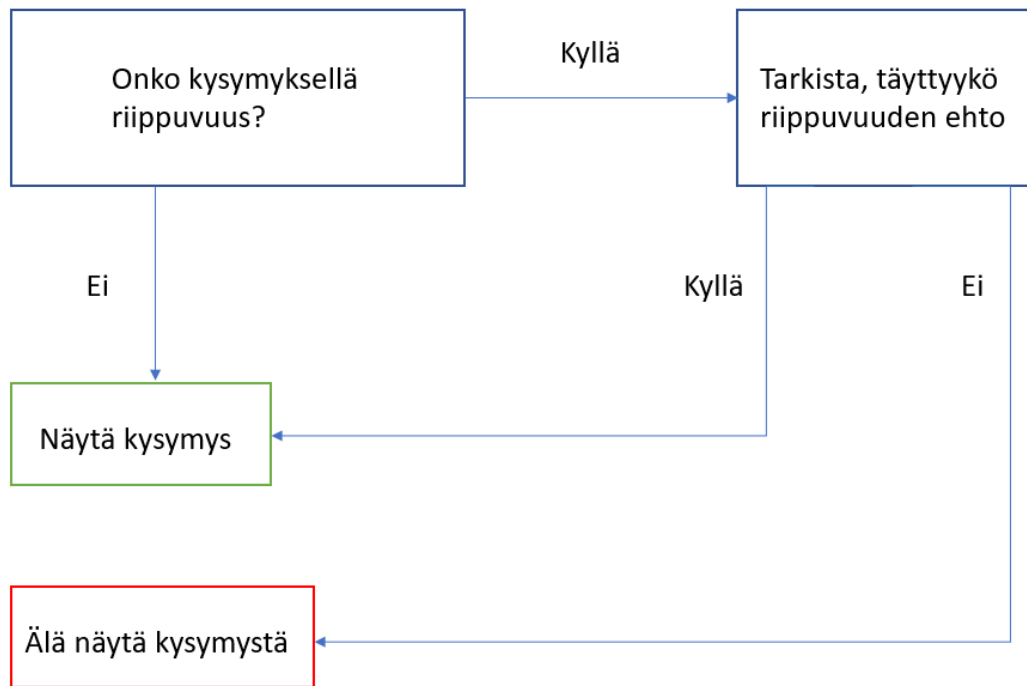
Käyttäjällä voi valita tilattavat tuotteet asiakkaalle, jos hän klikkaa tuotteen nimeä, ja täyttää tuotteen alapuolella olevat input-kentät tilaajan tiedoilla. Tilauksia voi

tehdä useammankin yhdelle tuotteelle, jos yhteystieto haluaa esimerkiksi tilata myös kaverilleen kyseisen tuotteen. Uuden tilaajan lisääminen tapahtuu painamalla plus-merkkiä tuotteen input-kenttien alapuolella.

Kysymykset ja vastaukset -osiossa käyttäjä voi nähdä myyntikampanjaan kuuluvat kysymykset. Vastaukset näkyvät kysymystekstin alapuolella. Jos käyttäjälle on aiemmin soitettu ja hän on vastannut kysymyksiin, vastaukset näkyvät valmiiksi täytettyinä. Kysymystyyppejä on kolme erilaista: vapaateksti-, monivalinta- ja valintakysymys. Vapaatekstikysymyksessä on tekstikenttä, johon voi vapaasti kirjoittaa mitä tahansa tekstiä. Monivalintakysymyksessä käyttäjä voi valita ennalta määritetyistä vastausvaihtoehdoista useamman vaihtoehdon. Valintakysymyksessä käyttäjä voi valita ainoastaan yhden vaihtoehdon ennalta määritetyistä vastausvaihtoehdoista.

Kysymyksiä ei tarvitse aina näyttää, koska kysymyksiä voi olla paljon. Jos kaikki kysymykset listattaisiin, ne täyttäisivät sivun ja se tekisivät siitä vaikealukuisen. Projektin määritelmässä oli myös, että Management-näkymässä voisi lisätä riippuvuuden kysymykselle. Jos riippuvuus täyttyy, niin kysymys näytetään sivulla. Jos riippuvuus ei täyty, niin kysymystä ei näytetä sivulla.

Kysymyksellä voi olla kolme erilaista riippuvuutta, jonka perusteella se näytetään: Kysymysriippuvuus, tuoteriippuvuus ja kontaktiriippuvuus. Kysymysriippuvuus tarkoittaa sitä, että kysymys on riippuvainen toisesta kysymyksestä ja sen vastauksista. Tuoteriippuvuus tarkoittaa sitä, että kysymys on riippuvainen jostain tietystä tuotteesta. Jos tuote on valittu, näytetään kysymys. Kontaktiriippuvuus tarkoittaa sitä, että kysymys on riippuvainen jostain kontaktin tietojen arvosta, esimerkiksi postinumerosta. Jos kontaktin tieto täyttää määritellyn ehdon, niin näytetään kysymys. Kuvio 5 näyttää, miten kysymys näytetään sivulla.



KUVIO 5. Vuokaavio kysymyksen näyttämisestä sivulla.

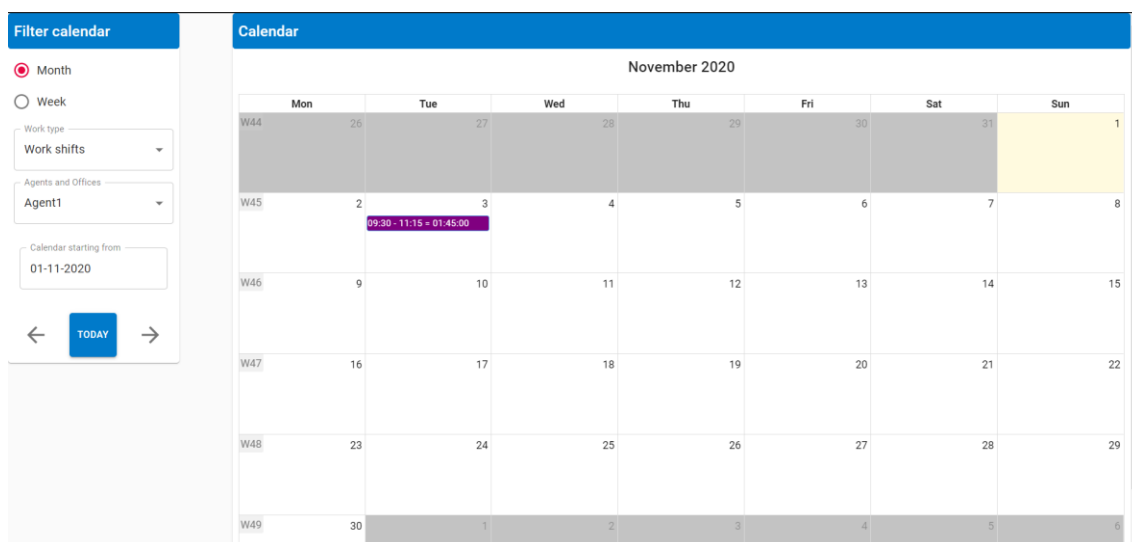
#### 6.4 Lopputuloksen kirjaaminen

Lopputulos kirjataan aina myyntitapahtuman päätteeksi. Myyntitapahtuma ei voi päättyä ilman lopputulosta. Lopputulos voi olla esimerkiksi: "ei vastaa" tai "tuote-tilaus". Lopputuloksen kirjaaminen tapahtuu kontakti-sivulla. Sivun oikealla reualla on lopputuloslista, josta käyttäjä valitsee tilanteeseen sopivan lopputuloksen.

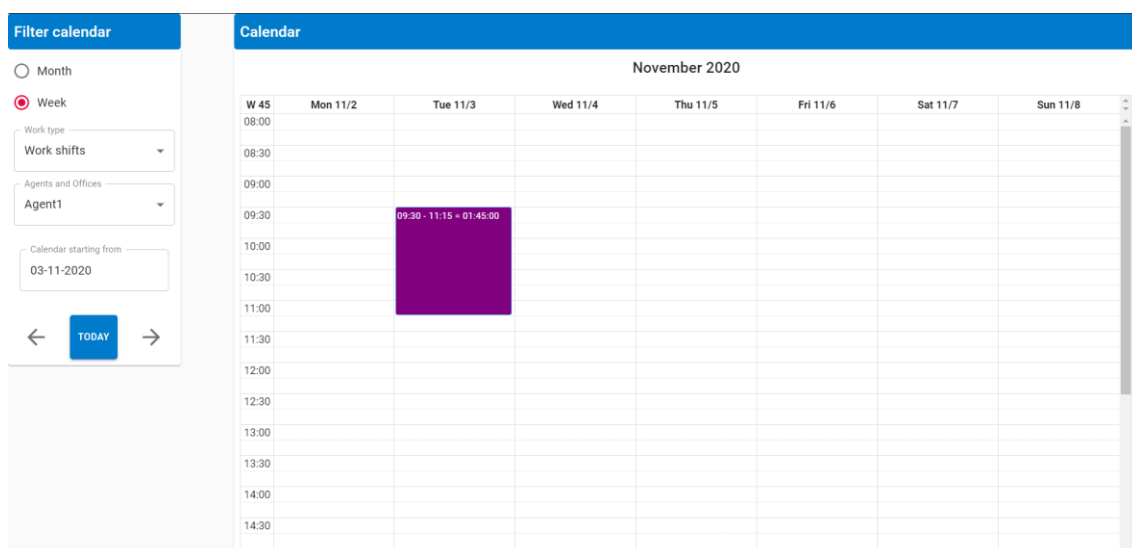
Lopputuloksessa voi olla Management-käyttöliittymän kautta määritelty tilaus- tai kysymystarkistus. Jos käyttäjän valitsemassa lopputuloksessa on tilaus- tai kysymystarkistus, niin applikaatio suorittaa tarkistuksen. Tilaustarkistuksessa tarkastetaan, että tuotteita on valittu vähintään yksi. Kysymystarkistuksessa tarkistetaan, että onko pakollisiin kysymyksiin vastattu. Jos tarkistus ei mene läpi, tulee käyttäjälle yläpalkkiin virheilmoitus, eikä lopputulosta kirjata. Jos tarkistus menee läpi, niin tapahtuman tiedot kirjataan palvelimelle, ja käyttäjä voi sulkea Agent-käyttöliittymän ICC managerista, ja sen jälkeen vastaanottaa uusia puheluita.

## 6.5 Kalenteri-sivu

Kalenteri-sivulla käyttäjä voi tarkastella omia työvuorojansa, työaikaansa, sovitujen uusintasovitojen ajankohtia sekä sisäänkirjautumisia. Kalenteri-sivu on sama mikä on käytössä Management-näkymässä, mutta pieniä eroja löytyy. Suurin ero on se, että käyttäjä voi ainoastaan lisätä työvuorojaan, mutta ei poistaa niitä. Käyttäjä näkee vain omat työvuoronsa, eikä kaikkien työntekijöiden. Kalenteriin pääsee ICC managerin yläpalkissa olevan valikon kautta. Kalenteri aukeaa ICC managerin sisällä olevaan Chromium-ikkunaan, samalla tavalla kuin Agent-käyttöliittymä. Kuva 17 ja kuva 18 näyttää kuukausinäkömän ja viikonnäkömän kalenterissa.



KUVA 17. Kalenteri-näkymän kuukausinäkömä.



KUVA 18. Kalenteri-näkymän viikonnäkömä.

## 7 TULOS

Lopputuloksena syntyi kaksi käyttöliittymää, jotka sisälsivät suunnitellut ominaisuudet ja ne täyttivät asiakkaan vaatimukset. Verkkosovellukset valmistuivat määräajassa, ja asiakas otti palvelut käyttöön onnistuneesti. Käyttöliittymien kehitystyö aloitettiin heinäkuussa 2020, ja sovellukset luovutettiin asiakkaalle lokakuussa 2020. Käyttöliittymien uudet kehitystyöt sovitaan asiakkaan kanssa ja ylläpito on Visma AI Worksin vastuulla.

### 7.1 Management-käyttöliittymä

Management-käyttöliittymä ei ole täysin valmis määräajan päättyessä. Asiakkaan testauksessa kävi ilmi, että käyttöliittymien tietyt ominaisuudet eivät olleet valmiita käyttöönottoa varten. Tämän takia asiakas päätti viivästyttää käyttöönottoa. Tarvittavat toiminallisuudet olivat molempien osapuolen takia jäänyt huomioimatta, joka aiheutti myöhästymistä. Asiakkaalta tuli myös määräys aikataulun nopeuttamisesta, jonka takia jouduimme priorisoimaan käyttöönottoa muiden ominaisuuksien yli. Asiakas antoi positiivista palautetta käyttöliittymän ulkonäöstä ja responsiivisuudesta. Tuote oli helppo kehittää, mutta vaati paljon aikaa.

Jatkokehitystä jatketaan Visma AI Worksin ja asiakkaan tahdosta, jota varten kehitetään jatkosuunnitelma. Suunnitelmassa aiotaan priorisoida tarvittavat ominaisuudet, jotta asiakas saa tuotteen käyttöön mahdollisemman nopeasti.

### 7.2 Agent-näkymä

Agent-käyttöliittymä saatiin valmiiksi määräajassa, ja asiakas testasi palvelua runsaasti ennen lopullista käyttöönottoa. Asiakkaan palautteen perusteella tehtiin muokkauksia käyttöliittymään ulkonäköön tai toiminnallisuuteen, jotka näkyivät julkaistussa versiossa. Tavoitteena oli saada mahdollisimman helppokäyttöinen käyttöliittymä, joka on myös responsiivinen. Agent-käyttöliittymästä tuli responsiivinen verkkokäyttöliittymä, joka sisältää paljon toiminnallisuutta. Vaikein vaihe kehityksessä oli kysymysten riippuvuuden lisääminen, ja se vei suurimman osan kehitystyön ajasta Agent-käyttöliittymän suhteen.

Jatkokehitys on Visma AI Worksin vastuulla, ja sitä tehdään aktiivisesti. Asiakas on halukas jatkamaan tuotteen kehitystä ja parannusta. Ominaisuuksia, joita ei ehditty tekemään ensimmäiseen versioon, tehdään jatkokehityksessä. Agent-käyttöliittymän osalta puuttuvia ominaisuuksia oli vähän. Suurin osa puuttuvista ominaisuuksista tuli selville asiakkaan palautteesta myöhemmin kehitystyön aikana, jolloin niitä ei ehditty lisäämään palveluun ennen ensimmäisen version julkaisua. Asiakas oli silti tyytyväinen palveluun ja kiitosta tuli käyttöliittymän yksinkertaisuudesta ja helppokäyttöisyydestä.

## 8 POHDINTA

Verkkokäyttöliittymien kehitys oli hektistä ja kehitystyö vaati suuren ponnistuksen kaikilta, jotta saisimme sen maaliin. Aikataulusta jouduttiin ottamaan noin neljä kuukautta pois johtuen ulkopuolisista syistä. Tämän lisäksi työmäärä käyttöliittymien kehityksen osalta kasvoi projektin edetessä. Asiakaspalavereissa tuli ilmi uusia kehityspyyntöjä käyttöliittymiin, joita emme olleet osanneet ottaa huomioon projektin alussa asiakkaan kanssa. Verkkokäyttöliittymien käyttämä ohjelmointirajapinta hidasti kehitystyötä projektin loppuvaiheessa. Aika-arvioista saatiin pidettyä kiinni, ja asiakas testasi käyttöliittymiä runsaasti ennen käyttöönottoa. Osa asiakkaan pyytämistä kehitystöistä siirrettiin tulevaisuuden kehityskohteiksi, jonka jälkeen priorisoimme kehitystöitä, että saisimme valmiin palvelun asiakkaalle määräaikaan mennessä.

Osa käyttöliittymien ominaisuuksista jouduttiin siirtämään tulevaisuuden kehityskohteiksi johtuen ajallisista syistä. Jatkokehitys tapahtuu Visma AI Worksin toimesta, ja uusista sekä siirretyistä kehitystöistä sovitaan asiakkaan kanssa erikseen. Tärkein jatkokehityskohde Management-käyttöliittymän osalta on Raportointi-näkymän valmistuminen. Asiakas tarvitsee Raportointi-näkymän dataa, jotta asiakas voi nähdä kaikki vaadittavat tiedot myyntikampanjoista. Agent-käyttöliittymän osalta jatkokehitystöitä ei kirjoittamisen hetkellä ole. Tulevaisuudessa käyttöliittymien ulkonäkö muokataan Visma AI Worksin tyyliohjeiden mukaiseksi.

Projekti oli mielestämme onnistunut. Asiakas otti käyttöön kehittämämme verkkokäyttöliittymät onnistuneesti, sekä saimme projektin valmiiksi määräaikaan mennessä, vaikka aikataulu oli hyvin tiukka. Kommunikaatio toimi hyvin tiimin sisäisesti, sekä opimme molemmat Reactista sekä siihen liittyvistä teknologioista. TypeScript helpotti suuresti kehitystyötä.

## LÄHTEET

Abramov, D. 2019. React v16.8: The One With Hooks. <https://reactjs.org/blog/2019/02/06/react-v16.8.0.html>

Dawson, C. 2014. JavaScript's History and How it Led To ReactJS. <https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/>

Flanagan, D. 2020. JavaScript: The Definitive Guide, 7th Edition. California: O'Reilly Media, Inc.

Kenton, W. 2020. Telemarketing. Luettu 16.9.2020. <https://www.investopedia.com/terms/t/telemarketing.asp>

Mardan, A. 2017. React quickly: painless web apps with React, JSX, Redux, and GraphQL. New York: Manning Publications Co.

Mohan, M ja Prusty, N. 2018. Learn ECMAScript – Second Edition. Packt Publishing

Mozilla ja yksittäiset avustajat. 2020a. What is JavaScript? Luettu 17.9.2020. [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

Mozilla ja yksittäiset avustajat. 2020b. JavaScript. Luettu 17.9.2020. <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

Negrino, T. 2012. JavaScript: Visual QuickStart Guide, Eighth Edition. Beachpit Press

React. 2020. React - A JavaScript library for building user interfaces. <https://reactjs.org/>

React. 2020. Components and Props. <https://reactjs.org/docs/components-and-props.html>

React. 2020. Introducing JSX. <https://reactjs.org/docs/introducing-jsx.html>

React. 2020. Using the State Hook. <https://reactjs.org/docs/hooks-state.html>

React. 2020. Using the Effect Hook. <https://reactjs.org/docs/hooks-effect.html>

Redux. 2020. A Predictable State Container for JS Apps. <https://redux.js.org/>

Vanderkam, D. 2019. Effective TypeScript. California: O'Reilly Media, Inc.

Vähä-Ruka, E. 2015. Mitä on outbound-markkinointi? Luettu 16.9.2020. <https://www.powermarkkinointi.com/blogi/miten-inbound-markkinointi-eroaa-outbound-markkinoinnista>