

Rane Hyvönen

# Virtuaalisten avustajien hyödyntäminen web- käyttöliittymissä



Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2023



KAMK • University  
of Applied Sciences

## Tiivistelmä

**Author:** Rane Hyvönen

**Työn nimi:** Virtuaalisten avustajien hyödyntäminen web-käyttöliittymissä

**Tutkintonimike:** Insinööri (AMK), tieto- ja viestintätekniikka

**Asiasanat:** virtuaalinen avustaja, kojelauta, ohjelmistokehitys, käyttöliittymä

**Kansikuva:** Freepik – pch.vector

Virtuaalisten avustajien suosio ja käyttö ovat olleet kasvussa jo jonkin aikaa. Suurin syy tähän on ollut AI-tekniikan kehitys ja integrointi virtuaalisiin avustajiin. Tästä huolimatta virtuaaliset avustajat eivät vielä ole saaneet suurta jalansijaa käyttöliittymäsovelluksissa. Opinnäytetyön päätavoitteena oli integroida virtuaalinen avustaja webpohjaiseen käyttöliittymään siten, että se ei harhauta tai häiritse käyttäjää. Toinen tavoite oli tehdä työ täysin avoimeksi, eli taulukot ja itse koodi ovat täysin jokaisen nähtävissä ja käytettävissä.

Opinnäytetyö jaettiin neljään eri osaan. Ensimmäisessä osassa käytiin läpi, mitä virtuaaliset avustajat ovat, mitkä ovat virtuaalisen avustajan tunnusmerkit ja mitä eroa on virtuaalisella avustajalla ja älykkäällä virtuaalisella avustajalla. Toisessa osassa esiteltiin käyttöliittymä, jonka pohjalle virtuaalinen avustaja rakennetaan. Kolmas osa käsitteli virtuaalisen avustajan toimivuuden, ulkonäön, rajoitusten ja sääntöjen suunnittelun. Viimeinen osuus keskittyi virtuaalisen avustajan toteuttamiseen.

Työssä kehitettiin yksinkertainen virtuaalinen avustaja, joka on osana webpohjaista käyttöliittymää. Avustaja kykenee esittämään käyttäjälle ehdotuksia käyttäjäkohtaisesti, mitä tehdä käyttöliittymä-tietojen perusteella tai viestien mukaan. Tämän lisäksi virtuaalinen avustaja lajittelee tärkeät tiedot ensin, minkä jälkeen se tarjoaa ne työntekijän näkökulmasta psykologisesti positiivisessa järjestyksessä, mikäli se on mahdollista. Työ on myös tavoitteiden mukaisesti täysin avoin, joka antaa muille mahdollisuuden edetä työssä eteenpäin tulevaisuudessa tai tarjota pohja muille kehittäjille itse työnä tai inspiraationa.

Työ demonstroi kolme eri asiaa. Virtuaalisten avustajien integrointi käyttöliittymiin on mahdollista häiritsemättä tai harhauttamatta käyttäjää. Virtuaalinen avustaja voi auttaa käyttäjiä tekemään parempia valintoja. Ei ole olemassa yhtä avustajaa, joka kävisi jokaiseen käyttötapaukseen. Eli jokaista avustajaa tulee muokata käyttökohteen vaatimuksien ja tarpeitten mukaisesti.

## Abstract

**Author:** Rane Hyvönen

**Title of the Publication:** Utilization of Virtual Assistants in Dashboards

**Degree Title:** Bachelor of Engineering, Information and Communication Technology

**Keywords:** virtual assistant, dashboard, software development, reactions, emotional control

**Cover:** Freepik – pch.vector

The popularity and adaptation of virtual assistants has been on the rise for some time. A big reason for this increase has been the improvements in the AI technology and how it has been integrated with the virtual assistants. Despite this, there has not been any large adoption of virtual assistants in the dashboard software sector. The main goal of this thesis was to integrate a virtual assistant to a web-based dashboard in a way that would not distract or bother the end-user. The second goal was to make the whole thesis open source from the code to the sheets that are used.

The thesis is divided to four different parts. The first part explains what virtual assistants are and what makes a thing virtual assistant and how it differs from intelligent virtual assistants. The second part discusses the rundown of the dashboard, the base in which the virtual assistant is built on. The third part goes over the planning of the virtual assistant functions, how it looks, limits, rules, and how it should affect the user or the worker. The last part focuses on creation of the virtual assistant.

In the thesis, a simple virtual assistant was developed, that is part of a web-based dashboard. The assistant is able to present suggestions to the user on user-by-user bases and are based on information of the dashboard or the messages. In addition, the assistant will sort the suggestions first by the importance and then by what affects the user positively psychologically, if that is possible. As per the goals, the whole thesis is open source which gives others an opportunity to further expand the thesis in the future or give an option to other developers to use the thesis as a base for a project or take inspiration from it.

The thesis demonstrates three different things. Virtual assistant integration is possible without distracting or bothering the end-user. Virtual assistant can help users make better decisions. Lastly, there is no one virtual assistant that works for every use case. This means every virtual assistant needs to be adjusted based on the requirements and the needs of the use case.

## Sisällys

1	Johdanto .....	1
2	Virtuaalinen avustaja .....	3
2.1	Virtuaalisten avustajien historia.....	3
2.2	Virtuaaliavustajien eroavaisuudet.....	5
2.3	Virtuaalisen avustajan käyttötarkoitukset .....	6
3	Työn lähtötilanne.....	7
3.1	Tietokanta .....	7
3.2	Palvelin .....	8
3.3	Selainpalvelin.....	9
3.4	Ulkonäkö ja muotoilu .....	10
3.5	Ominaisuudet ja toiminta.....	11
4	Virtuaalisen avustajan vaatimusmäärittely ja suunnittelu .....	13
4.1	Ulkoasu.....	14
4.2	Teknisten vaatimusten toteutus .....	16
5	Virtuaalisen avustajan toteuttaminen.....	18
5.1	Ulkoasu.....	18
5.2	Tietokanta .....	19
5.3	Palvelin .....	19
5.4	Selainpalvelin ja VA-priorisointi .....	22
6	Yhteenveto .....	25
7	Pohdinta .....	26
	Lähteet .....	28
8	Liitteet.....	31

## Symboliluettelo

IVA	Intelligent virtual assistant (älykäs virtuaalinen avustaja) eli virtuaalinen avustaja, joka käyttää tekoälyä [1].
AI	Artificial intelligence (tekoäly) on koneen kyky näyttää ihmisen kaltaisia ominaisuuksia, kuten päättelyä, oppimista, suunnittelua ja luovuutta [2].
VA	Virtual Assistant (virtuaalinen avustaja) on teknologia, joka on suunniteltu autamaan käyttäjää vastaamalla kysymyksiin ja käsittelemällä tehtäviä [3].
API	Application programming interface (ohjelmointirajapinta) on tapa kommunikoida tietyn tietokoneohjelman tai Internet-palvelun kanssa [4].
HTTP	Hypertext transfer protocol (hypertekstin siirtoprotokolla) on tietokoneohjelman tekemät ohjeet, joiden avulla tietokoneesi voi muodostaa yhteyden Internet-asiakirjaan [5].
RTE	Runtime environment (suoritusympäristö) on ympäristö, jossa ohjelma tai sovellus suoritetaan. Se on laitteisto- ja ohjelmistoinfrastruktuuri, joka tukee tietyn koodikannan käyttöä reaaliajassa [6].
JSON	JavaScript Object Notation on kevyt tiedonsiirtomuoto [7].
CSS	Cascading Style Sheets (porrastetut tyyliarkit) on tietokoneohjelmisto, jonka avulla voi kuvata ja hallita, miltä verkkosivun tai verkkosivujen sarjan tulee näyttää käyttäjälle [8].

## 1 Johdanto

Virtuaaliset avustajat (VA) ja älykkäät virtuaaliset avustajat (IVA) ovat merkittävä osa nykypäivän teknologiaa, internetiä ja asiakaspalvelua, ja niitä hyödynnetään mittavasti palveluiden automatisoinnissa ja digitalisoinnissa. Virtuaalinen avustaja on teknologia, joka on suunniteltu auttamaan käyttäjää vastaamalla kysymyksiin ja käsittelemällä tehtäviä [3]. Esimerkkejä IVA-ohjelmista ovat Applen älypuhelimista löytyvä Siri [9] ja useilla asiakaspalvelusivustoilla käytettävät chatbotit. IVA- ja VA-ohjelmat voivat säästää yrityksiltä paljon rahaa ja vaivaa asiakaspalvelussa ja voivat auttaa jokaista arkielämässä ja työtehtävissä.

Virtuaalisten avustajien sovelluskohteiden määrä voi suurentua tai jopa räjähtää eksponentiaalisesti. Syynä tähän on niiden saama laaja julkisuus, suuren yleisön mielenkiinnon herääminen sekä yleisen ymmärryksen lisääntyminen tekoälyn mahdollisuuksista. Hyvä esimerkki tästä on Open AI -yrityksen vuonna 2022 julkaisema ChatGPT-sovellus, joka pystyy usein antamaan oikeita ja personalisoituja vastauksia kysymyksiin tai tehtäviin. Bloomberg-mediayhtiön journalisti Vlad Savov nimesi jopa yhden hänen artikkeleistaan otsikolla ”ChatGPT voi olla AI:n iPhone-hetki” [10].

Kaikesta tästä huolimatta on vaikea löytää käyttöliittymäFituotetta, joka sisältäisi sellaisen virtuaalisen avustajan, joka kykenee auttamaan käyttäjää valintojen tekemisessä käyttöliittymän tuottamien tietojen perusteella. Tämä ongelma loi idean ja tavoitteen opinnäytetyölle. Tavoitteena on selvittää, mitä IVA- ja VA-ohjelman tekeminen vaatii ja miten VA-ominaisuus tai -ohjelma implementoidaan olemassa olevaan web-käyttöliittymään.

Opinnäytetyön aihe ja kehittämistyössä hyödynnetty käyttöliittymän pohja liittyvät kirjoittajan INTK:lla suoritettuihin työtehtäviin ja tutkimuksiin. INTK on hollantilainen internet- ja markkinointipalveluja tarjoava yritys, joka on antanut luvan käyttää pohjaa avoimena lähdekoodina tätä työtä varten. Lähdekoodit ovat opinnäytetyön tekijän tuottamia, ja työssä käytetyn tietokannan tietoja on muokattu yksityisyyden takaamiseksi.

Työn pohjana toimii web-käyttöliittymä (engl. web dashboard), joka koostuu selainpalvelusta nimeltään React, palvelinpuolesta NodeJS ja Google Sheets -tietokannasta. Toteutusideana on käyttää edellä mainittuja teknologioita ja lisätä VA-pohjainen huomautus- ja ohjaustoiminto web-näkymään.

Työn teoriaosuus on jaettu kolmeen osaan. Ensimmäisessä osassa tarkastellaan virtuaalista avustajaa ja sen eroavaisuuksia älykkääseen virtuaaliseen avustajaan. Toisessa osuudessa käydään läpi

työn lähtötilanne, eli miten sovellus toimi ennen työn toteutusta. Viimeisessä osuudessa tarkastellaan realistista tavoitetta ja toteutukseen liittyviä esteitä ja haasteita. Viimeisessä osassa käydään läpi työn suunnitteluprosessi ja valinnat vaatimuksien mukaan. Tämän lisäksi viimeinen osa sisältää työn tulosten tarkastelun ja perustelut muutoksille, jotka vaikuttivat työn lopputulokseen.

## 2 Virtuaalinen avustaja

Yleisesti virtuaalinen avustaja (VA) on teknologia, joka on suunniteltu auttamaan käyttäjää vastaamalla kysymyksiin ja käsittelemällä tehtäviä [3]. Tämä voi tapahtua eri muodoissa, kuten tehtävien suorittamisen opastuksena, mutta myös vastausten antamisena käyttäjälle joko keskustelun tai muun tiedon perusteella. Kyseinen tieto voi tulla joko ulkoisesta lähteestä, kuten internetistä, tai sisäisestä lähteestä, kuten lokaalisesta tietokanasta tai avustajaan suoraan ohjelmoiduista tiedoista.

### 2.1 Virtuaalisten avustajien historia

Virtuaalinen avustaja ei ole uusi keksintö. Virtuaaliset avustajat ovat olleet kehityksen kohteena vuosikymmeniä. Esimerkiksi IBM Shoebox keksittiin vuonna 1961. Tämä laite pystyi kuuntelemaan käyttäjää ja vastaamaan 16:een eri sanaan. Käyttäen näitä sanoja laite pystyi laskemaan yksinkertaisia lisäys- ja vähennyslaskuja. [11].

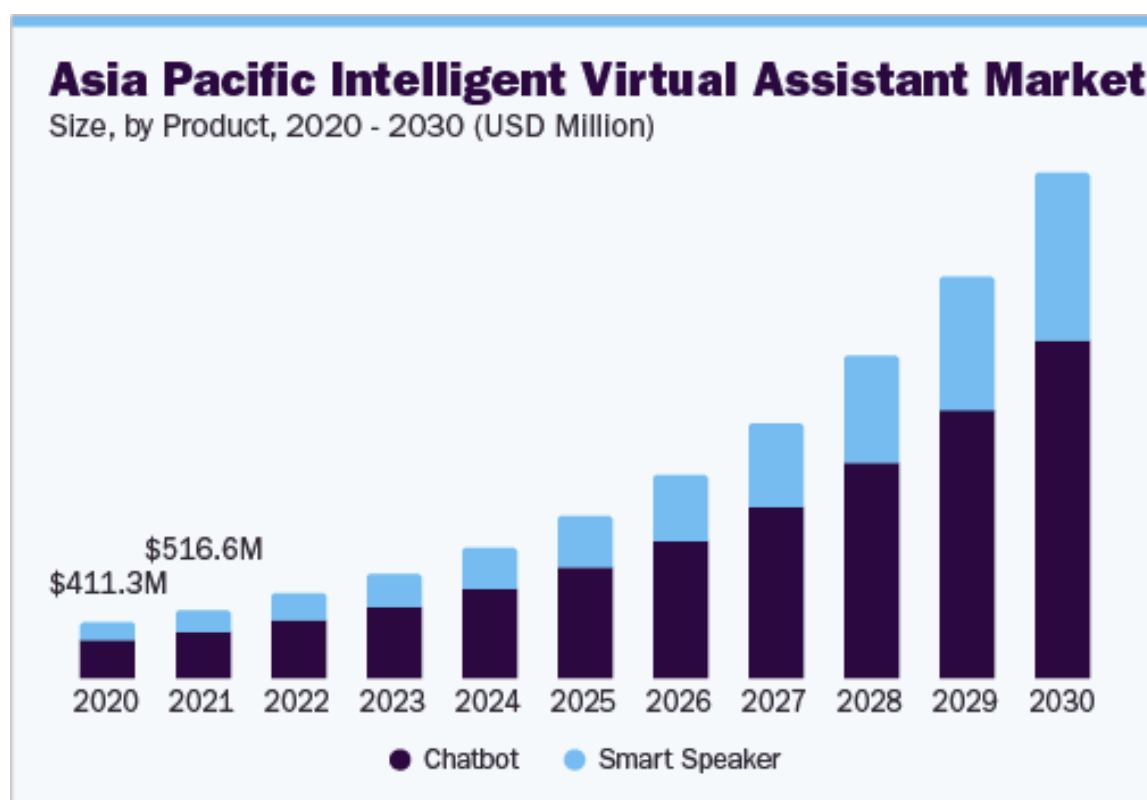
Vaikka virtuaalisten avustajien juuret eivät ole niin syvät kuin monen muun teknologian, VA-teknologia on levinnyt nopeasti. Noin 50 vuotta ensimmäisen virtuaalisen avustajan julkaisusta yleiset virtuaaliset assistentit tulivat suurten massojen käyttöön Applen Siri iPhone -ominaisuuden ansiosta. Siri pystyi hakemaan tietoa internetistä ja suorittamaan erinäisiä tehtäviä, kuten asettamaan herätyksen tai soittamaan pyydettyyn puhelinnumeroon. Tämä oli suuri muutos verrattuna laajasti tunnettuun virtuaaliseen avustajaan Microsoft Clippyyn, joka on Microsoft Word -tuotteen toiminnallisuus. Suurin syy, miksi Siri tuntui niin suurelta teknologiahypyltä, lienee se, että Clippy pystyi avustamaan vain tehtävissä Word-sovelluksen sisällä.

Siristä eteenpäin virtuaaliset avustajat eivät olleet tehneet suuria muutoksia siihen, miten käyttäjät toimivat niiden kanssa. Käytännössä käyttäjä kysyy kysymyksen tai antaa käskyn teksti- tai puhemuodossa, ja avustaja vastaa tai suorittaa käskyn mukaisen tehtävän. Sen sijaan suurimmat muutokset olivat tuotteissa, jotka tarjosivat VA-toimintoja ja pyrkivät saamaan avustajat älykkäämmiksi ja muistuttamaan enemmän oikeaa ihmistä. Yksi varhaisimmista tuotteista, joka yhdistää nämä kaksi ideaa, oli Amazonin Echo ja Amazonin Alexa. Alexa on IVA-ohjelma, jota Echo käyttää avustaakseen käyttäjiä. Applen Siri pystyy suorittamaan samoja tehtäviä kuin Alexa, mutta Amazonin Echo myytiin smart home -tuotteena, jonka päätarkoituksena oli toimia virtuaalisena avustajana kotona toisin kuin Siri. Siri ja Alexa pystyvät molemmat etsimään tietoa



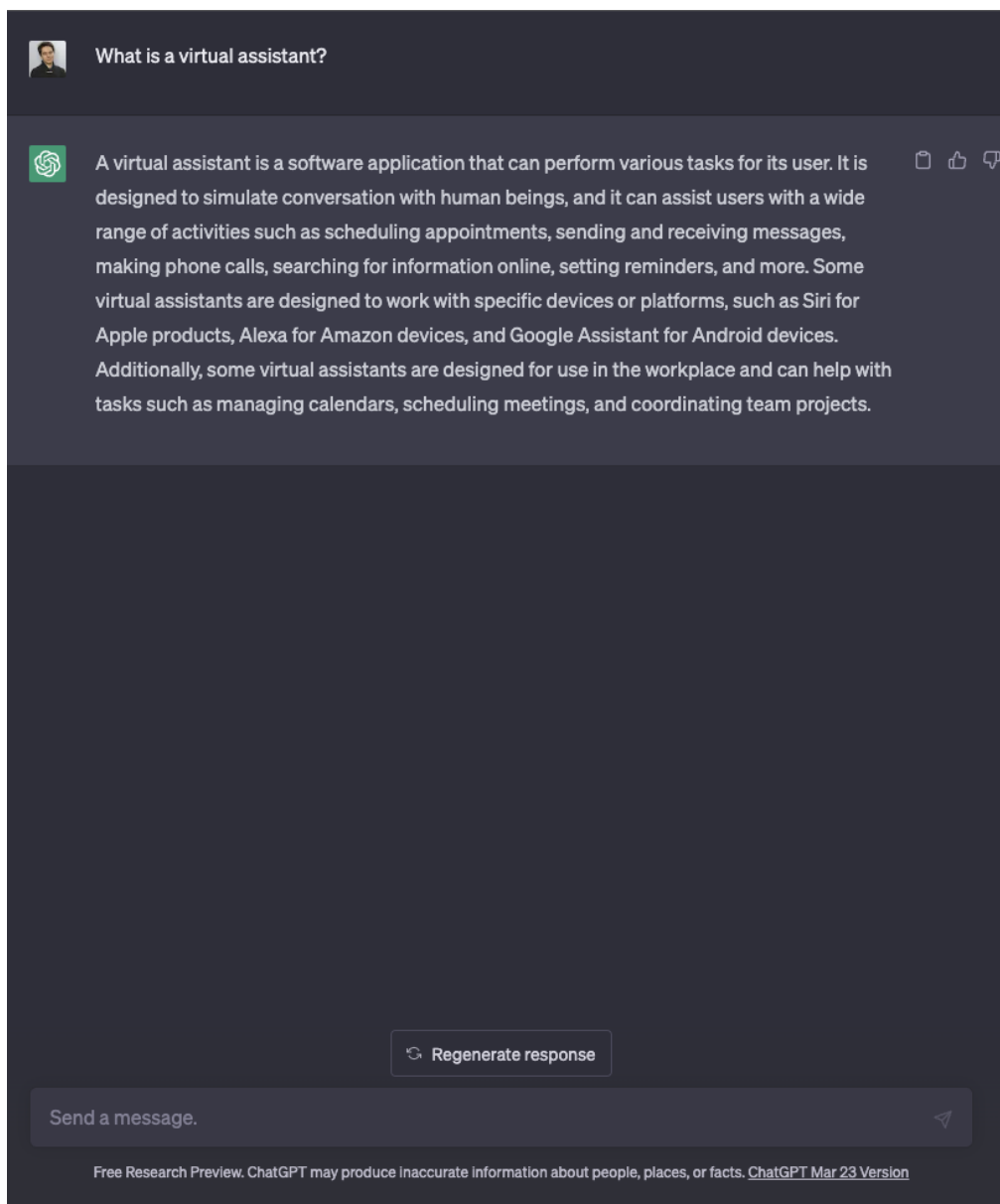
internetistä, asettamaan herätyksiä tai lähettämään tekstiviestejä, mutta Alexalla on Amazon-palveluihin liittyviä lisäominaisuuksia, kuten ostoksien lisääminen ostoskoriin tai ostaminen valittuun kotiosoitteeseen. [12.]

Vaikka virtuaaliset avustajat ovat saaneet paljon huomiota suurilta yrityksiltä, kuten Google, Alle ja Amazon, ihmisten kiinnostus VA-ratkaisuja kohtaan on pysynyt rajallisena. Tämä on nähtävissä kuvassa 1, jossa on esitetty Grand View Researchin arvio Aasian IVA-markkinasta vuosina 2023–2030.



Kuva 1. Arvio Aasian IVA-markkinasta vuosille 2023–2030 [13].

Kuvasta 2 havaitaan, että kiinnostus virtuaalisiin avustajiin on nousussa. Eräiden arvioiden mukaan virtuaalisten avustajien markkina tulee olemaan biljoonan dollarin arvoinen vuoteen 2025 mennessä [14]. Tämän arvion virtuaalisten avustajien markkinan noususta vahvistaa myös epäsuorasti OpenAI-yrityksen luoma ChatGPT-sovellus. Syy ChatGPT:n suureen nousuun on sen älykkyys ja hyödyllisyys, joka tukee ideaa ”Mitä älykkäämpi virtuaalinen avustaja tai sen käyttämä AI on, sitä hyödyllisempi se on käyttäjälle”. Työn kirjoitushetkellä ChatGPT pystyy vastaamaan kysymyksiin sekä luomaan sujuvaa tekstiä, kuten tarinoita ja koodia. Esimerkiksi kuvassa 2 kysytään ChatGPT:ltä, mikä on virtuaalinen avustaja.



Kuva 2. ChatGPT:n luoma selitys käsitteelle virtuaalinen avustaja.

## 2.2 Virtuaaliavustajien eroavaisuudet

Virtuaalisen avustajan ja älykkään virtuaalisen avustajan ero ei ole suuri: älykäs virtuaalinen avustaja sisältää AI-teknologiaa, kun taas perinteinen virtuaalinen avustaja ei sitä sisällä [1].

Monessa tapauksessa tämä ero ei ole helposti huomattavissa käyttäjälle, koska virtuaaliset avustajat ovat toiminnoiltaan pääpiirteissään samanlaisia. Ero on kuitenkin havaittavissa, kun avustajan toimintaa tutkitaan syvällisemmin. IVA vaikuttaa AI-teknologian ansiosta enemmän

ihmismäiseltä sen tuottaman puheen, tekstin tai toiminnan perusteella. Perinteinen virtuaalinen avustaja pystyy matkimaan ihmismäisyyttä, mutta hyvin rajoitetussa määrin. Tämä on helposti nähtävissä yleisimmissä keskustelu- ja avustusboteissa. Ne pystyvät tervehtimään kuin ihminen ja vastaamaan kysymyksiin, jotka on tarkasti määritelty tai haettavissa. Nämä vastaukset eivät kuitenkaan ole yhtä henkilökohtaisia, kuten älykkään virtuaalisen avustajan, vastaukset [1].

### 2.3 Virtuaalisen avustajan käyttötarkoitukset

Alun perin ajateltiin ja ennustettiin, että älykkäät virtuaaliset avustajat tulevat viemään suuren osa asiakaspalvelutöistä. Näin ei kuitenkaan käynyt vaan IVA- ja VA-ohjelmat ovat tulleet työntekijöiden avuksi [15]. Esimerkiksi Kindly, joka on AI-keskustelubottirakentaja, raportoi heidän AI-keskustelubottinsa hoitaneen 90 % Norwegian Block Exchange -yrityksen asiakaspalvelutapauksista. Tämä auttaa asiakaspalvelun ihmistiimiä keskittymään vaikeampiin ongelmiin [16]. Vaikka normaali virtuaalinen avustaja ei kykene vastaamaan niin moneen kysymykseen kuin AI-pohjainen keskustelubotti, se pystyy hoitamaan monet toistuvat ongelmat ja tilanteet. VA-keskustelubotti pystyy esimerkiksi vastaamaan yksinkertaisiin kysymyksiin tai ohjaamaan asiakkaan tietolähteisiin, jotka voisivat olla hyödyllisiä. Monessa tapauksessa tämä säästää suuren määrän rahaa ja ihmisresursseja yrityksissä.

Virtuaalisten avustajien sovellusalueita ovat muun muassa terveydenhuolto, jälleenmyynti, koulutus, toimitusketjut ja telekommunikaatio [17]. Nokian Digital Assistant on erinomainen esimerkki siitä, millaista työ voisi ideaalisesti olla tulevaisuudessa. Kyseinen avustaja on IVA-pohjainen ja koulutettu telekommunikointipohjaisiin tehtäviin. Avustaja pystyy auttamaan työntekijöitä etsimään tarvittavia tietoja, dokumentteja ja työkaluja nopeammin kuin perinteisin tavoin etsimällä [18].

### 3 Työn lähtötilanne

Toteutuksen pohja on rakennettu hyvin yksinkertaisella full stack -teknologiapinolla. Työ koostuu selainpalvelusta, palvelinpuolesta ja tietokannasta. Jokainen osa tekee oman osuutensa verkkosovelluksen luonnissa. Selainpalvelin tarjoaa käyttäjäystävällisen käyttöliittymän, jonka avulla käyttäjä pystyy olemaan vuorovaikutuksessa palvelinpuolen tiedon kanssa. [19.]

Palvelinpuoli hoitaa kolme eri tehtävää. Ensimmäinen tehtävä on tiedon hakeminen, jossa palvelin hakee tallennetut tiedot tietokannasta. Tämä voi tapahtua synkronoiduilla tai ajoituspohjaisilla tavoilla. Toinen tehtävä on muokata ja muuttaa tieto sellaiseen muotoon, joka toimii HTTP API -muodon kanssa. Viimeinen tehtävä on kommunikointi, eli vastata asianmukaisesti http-pyyntöihin. [19.]

Tietokanta on organisoitu kokoelma tietoa. Tietokannat ovat yleisesti helppoja muokata, vaikka tietojen määrä on suuri ja käyttäjiä on monia. Samalla tavalla ne ovat myös monesti hankalia työstää, jos osaamista ja kokemusta tietokannoista ei ole tai jos tietojen muokkaamiselle ei ole rakennettu käyttöliittymää. Tästä syystä moni käyttää laskentataulukoita tietokannan sijaan. Suurimmat erot tietokannan ja laskentataulukkojen välillä liittyvät tietojen muokkaamiseen ja tallentamiseen sekä siihen, kuka voi päästä tietoihin käsiksi ja kuinka paljon tietoa voidaan tallentaa. [20.]

#### 3.1 Tietokanta

Alkuperäinen ratkaisu hakee tiedot laskentataulukoista Google Sheets -tietokannasta. Tämä valinta luo helpon ympäristön kaikille työntekijöille, koska kaikki tieto on helposti saatavilla ja muokattavissa verkossa yksinkertaisessa taulukkomuodossa. Tämän lisäksi Google tarjoaa käyttäjille ja kehittäjille Google Sheets API -työkalun, jonka avulla palvelin pystyy lukemaan, kirjoittamaan ja muokkaamaan laskentataulukon tietoja [21]. Työn laskentataulukot sisältävät tietoja yrityksen työntekijöistä, laskuista ja asiakkaista. Työssä käytetyt esimerkkitaulukot on esitetty liitteessä 1.

Tietokannan käsittelyyn ulkoisesti tarvitaan kirjautumistiedot. Kyseinen vaihe voi tapahtua joko ihmisen tai sovelluksen pyynnöstä. Sovelluksen ja ihmisen kirjautuminen menee Googlen OAuth2-identifiointipalvelun kautta, mutta sovellukset ja palvelimet vaativat oman tilin Google-palveluiden käyttöön [22]. Kirjautumistietojen lisäksi palvelutilillä pitää olla oikeudet Google

Sheets API -ominaisuuksiin ja itse taulukoihin, jotta sovellus tai avustaja pystyy lukemaan ja muuttamaan taulukoita.

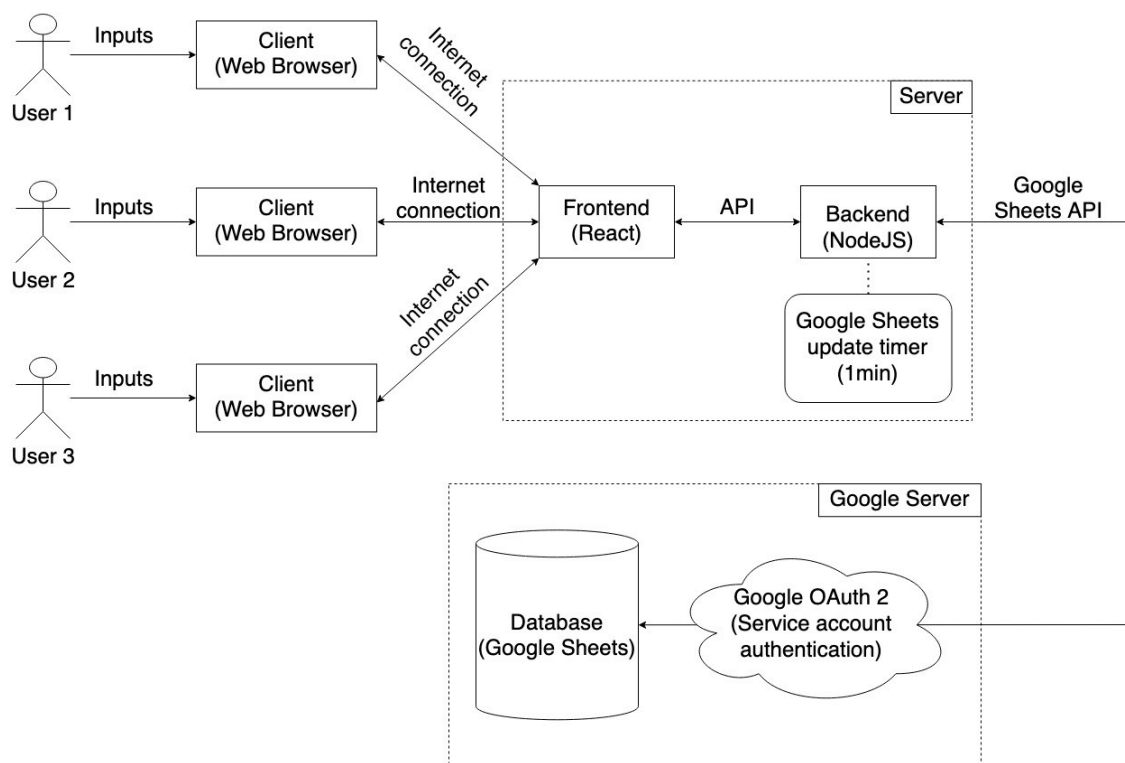
### 3.2 Palvelin

Palvelin on kokonaisuuden ydin, jonka avulla koko järjestelmä toimii ja kommunikoi muiden osien kanssa. Kyseinen osuus käyttää NodeJS-teknologiaa, jonka ohjelmointikieli on JavaScript [23].

Yksi palvelimen päätehtävistä on pyytää Google Sheets API:lta tietoa kriteerien mukaan. Nämä kriteerit ovat pääsytunnus, taulukko-id, sivun nimi ja tieto siitä, halutaanko taulukko lukea riveinä vai sarakkeina. Pääsytunnuksena toimii Googlen omat palvelutilin tunnukset, jotka saa Google Cloud Console -palvelusta. Jos tiedon hakeminen onnistui, tulee palvelimen muuttaa tämä tieto JSON-muotoon tiedon lähettämistä varten.

Tiedonhakuprosessin tulisi ideaalisesti tapahtua niin usein kuin mahdollista Google Sheets API:n säännöksien mukaan. Google Sheets API vaatii, että jokainen projekti pystyy lukemaan ja kirjoittamaan enintään 60 kertaa minuutissa per käyttäjä [24]. Järjestelmä päivittää Google Sheets -taulukoiden tiedot yhden minuutin välein. Tämä tapahtuu yksinkertaisella ohjelman sisäisellä ajastimella.

Palvelimen viimeinen tehtävä on vastata selainpalvelimen pyyntöihin. Nämä pyynnöt tapahtuvat http-protokollalla JSON-muodossa. Tämä mahdollistaa selainpalvelimen ja tietokannan sulavan viestinnän molempiin suuntiin. Tätä metodia käytettäessä ei ole mahdollista lähettää liikaa pyyntöjä Googlen API-palveluihin vahingossa. Etuna kuitenkin on se, että selainpalvelin saa tiedon nopeasti, koska käyttäjän ei tarvitse odottaa tiedon pyyntö- ja vastausprosessia Googlelta. Kuva 3 esittää palvelimen toiminta- ja kommunikaatioperiaatteen käyttäjien, Google-palveluiden ja identifioinnin kanssa.



Kuva 3. Palvelimen toimintaperiaate ja arkkitehtuuri

### 3.3 Selainpalvelin

Selainpalvelimen käyttöliittymä tarjoaa intuitiivisen ja interaktiivisen tavan käyttää virtuaalista avustajaa. Selainpalvelin perustuu Reactiin, joka on ilmainen ja avoimen lähdekoodin kirjasto JavaScriptille. React-projektin kehittäjä ja ylläpitäjä on Meta. React mahdollistaa interaktiivisten käyttöliittymien ja verkkosovellusten rakentamisen nopeasti ja tehokkaasti [25].

Koska selainpalvelimen React ja palvelimen NodeJS käyttävät JavaScriptiä, tulee kehittäjän ymmärtää JavaScript-ohjelmointia. JavaScript on kevyt ohjelmointikieli, joka kehitettiin vuonna 1995. Brendan Eich loi JavaScriptin NetScape 2 -verkkoselainta varten [26]. Nykyään JavaScript-ohjelmointikieltä käytetään verkkosivuissa, peleissä ja AI-sovelluksissa sekä selainpalvelimien ja palvelimien kehityksessä [27].

Työssä käyttöliittymän ulkonäön muokkaamiseen hyödynnetään Tailwind CSS -runkoa. Tailwind CSS on Reactin tapaan avoimen lähdekoodin projekti, jonka tarkoitus on helpottaa ja yksinkertaistaa käyttöliittymän ulkonäön muokkausta. Tailwind CSS eroaa monista muista CSS-

runkoprojekteista sillä, että se ei käytä valmiiksi annettuja komponentteja. Tämän sijaan Tailwind CSS tarjoaa luokkia, joista tekijä voi itse valita haluamansa komponentit [28].

### 3.4 Ulkonäkö ja muotoilu

Web-käyttöliittymän suurin hyöty on tiedon visualisoiminen. Tiedon tulee olla ymmärrettävissä helposti, loogisesti ja intuitiivisesti. Lisäksi tiedon määrän tulee olla rajoitettu, ja sen esittämisessä on suosittava yksinkertaisesti hahmotettavia tapoja. Hyvä esimerkki tästä on kuva 4, jossa on selkeä otsikko, viivojen selitteet ja akseleiden merkinnät. Vaaka-akselilla on esitetty aikajana kuukausina ja pystyakselilla liikevaihdon menetystä kuvaavat euromäärät.



Kuva 4. Menetetty liikevaihto kuukausitasolla vuosina 2021 ja 2022

Intuitiivisia ulkonäköideoita on mahdollista toteuttaa myös taulukoissa (kuva 5). Pohjavärit ovat hyvin neutraaleja, kuten harmaa ja valkoinen. Asiat, joihin käyttäjän halutaan kohdistavan huomiota, korostetaan neutraaleista väreistä erottuvilla tehosteväreillä. On silti tärkeää, että värit eivät ole liian voimakkaita ilman tarkoitusta. Punainen on erinomainen ja tärkeä väri web-suunnittelussa, koska se herättää käyttäjässä vaaran tai kiireen tunteen ja on näin ollen huomion keskipiste [29]. Punaisen värin tärkeys taulukoissa on suuri, koska taulukot sisältävät paljon tietoa.

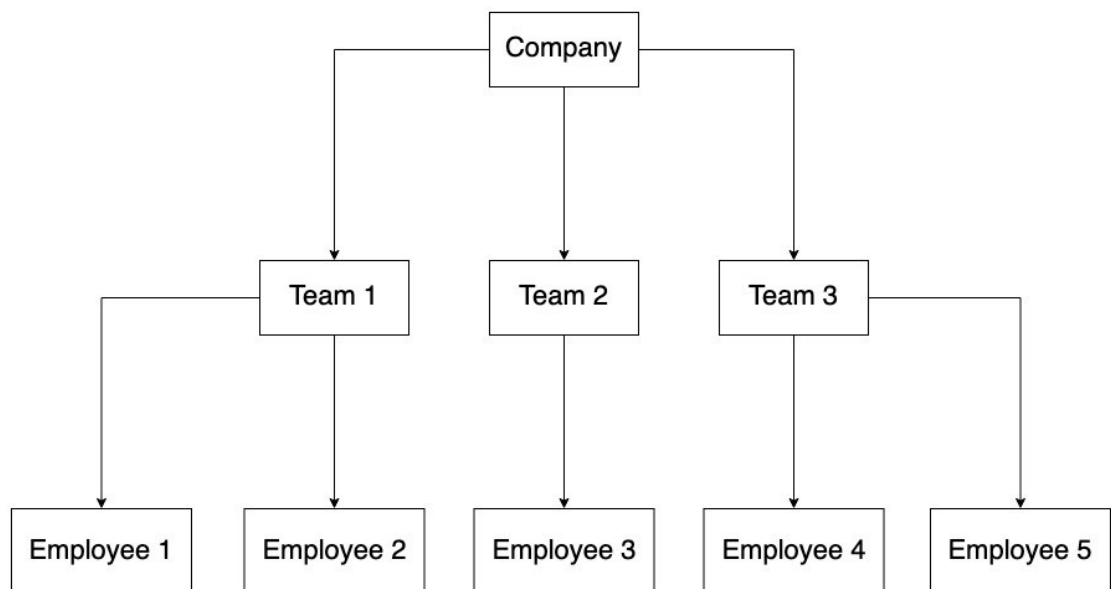
Taulukot yleisesti rikkovat tiedon määrää koskevia suosituksia, mutta värien avulla taulukoista voidaan korostaa käyttäjälle merkityksellistä tietoa.

Client 88	Worker 14	Netherlands	31-12-22	28-09-22	📈	Yes	8
Client 89	Worker 14	Netherlands	31-12-22		📈	Yes	8
Client 90	Worker 12	Netherlands	31-12-22	27-10-22	📈	Yes	8
Client 91	Worker 10	Netherlands	31-12-22	21-09-22	📈	Yes	8
Client 92	Worker 3	Spain	31-12-22		📈	Yes	8
Client 93	Worker 13	Belgium	31-12-22	03-11-22	📈	Yes	9
Client 94	Worker 8	Netherlands	31-12-22		📈	Yes	9

Kuva 5. Värien käyttöesimerkki taulukossa

### 3.5 Ominaisuudet ja toiminta

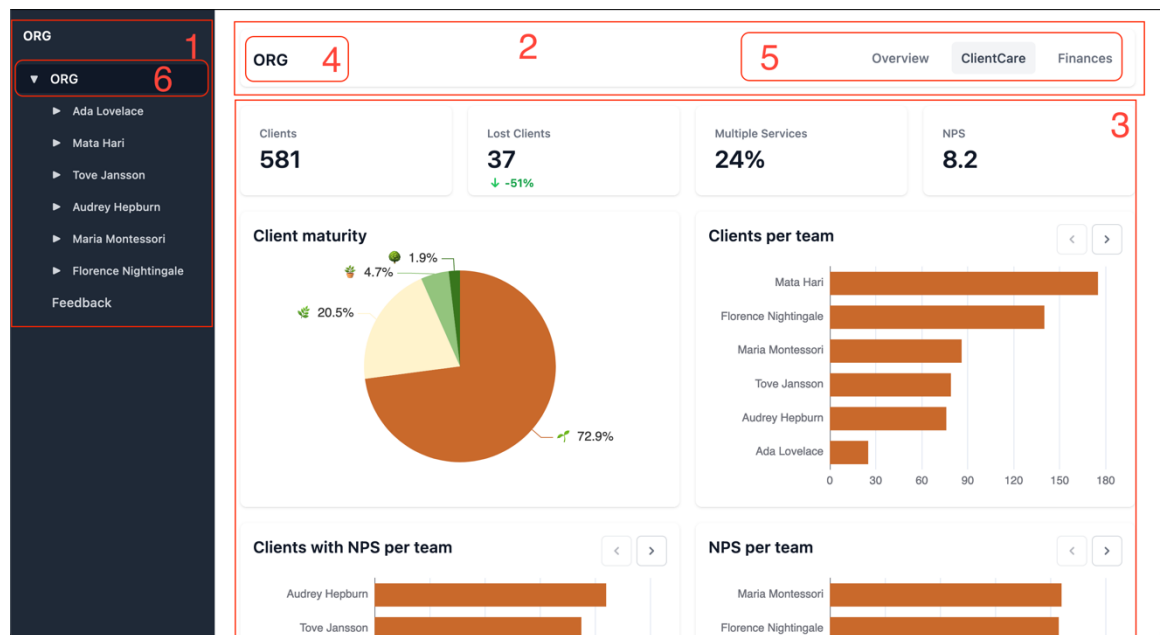
Käyttöliittymän suurin etu on se, että tietoa voidaan rajata halutusti. Tämä antaa käyttäjälle joustavuutta ja personalisointimahdollisuuksia. Yksi tärkeä ominaisuus on mahdollisuus eritellä ja yhdistää dataa eri näkökulmista. Itse toteutus tapahtuu yleiskatsausnäköymästä, jossa hierarkian tasoina ovat yritys, tiimit ja henkilöstö (kuva 6). Näillä eri näkömillä saadaan tilannetiedot sujuvasti esiin ja lasketaan samalla käyttäjiltä vaadittavan tietoteknisen osaamisen tasoa.



Kuva 6. Kaavio yleiskatsausnäköymän hierarkiasta



Lähtökohtana olleessa ratkaisussa tiedon rajaaminen on toteutettu sivupalkilla, jonka avulla näytön tila jaetaan näkymä- ja kategorianavigointiin ja sisältöosioon. Kuvassa 7 on nähtävissä selvä rajaslogiikka, miten eri navigointivalikot rajaavat sisältöä käyttäjän valintojen perusteella. Nämä tulevat esiin visuaalisilla avusteilla, kuten kuvaan merkityillä alueilla 5 ja 6. Näissä valitun asetuksen tausta on värjätty tummemmalla neutraalilla värillä. Sivuluettelolla (alue 1) avulla valitaan näkymäalue ja yläpalkista voidaan valita kategoria näkymälle (alue 5). Yläpalkki tarjoaa myös paikan otsikolle. Tämä on hyödyllinen keino tiedon välittämiseen mobiilinäkymässä, jossa käyttäjän sivupalkki on piilossa.



Kuva 7. Käyttöliittymän osioiden alkuperäinen asetelma

#### 4 Virtuaalisen avustajan vaatimusmäärittely ja suunnittelu

Työn lähtökohtana oleva ratkaisu mahdollistaa yksinkertaisen virtuaalisen avustajan integroinnin osaksi toimivaa järjestelmää. Virtuaalisen avustajan tavoitteena oli tarjota tukea muun muassa harjoittelijoiden ja työntekijöiden ohjaukseen. Virtuaaliselle avustajalle (VA) asetettiin seuraavat vaatimukset:

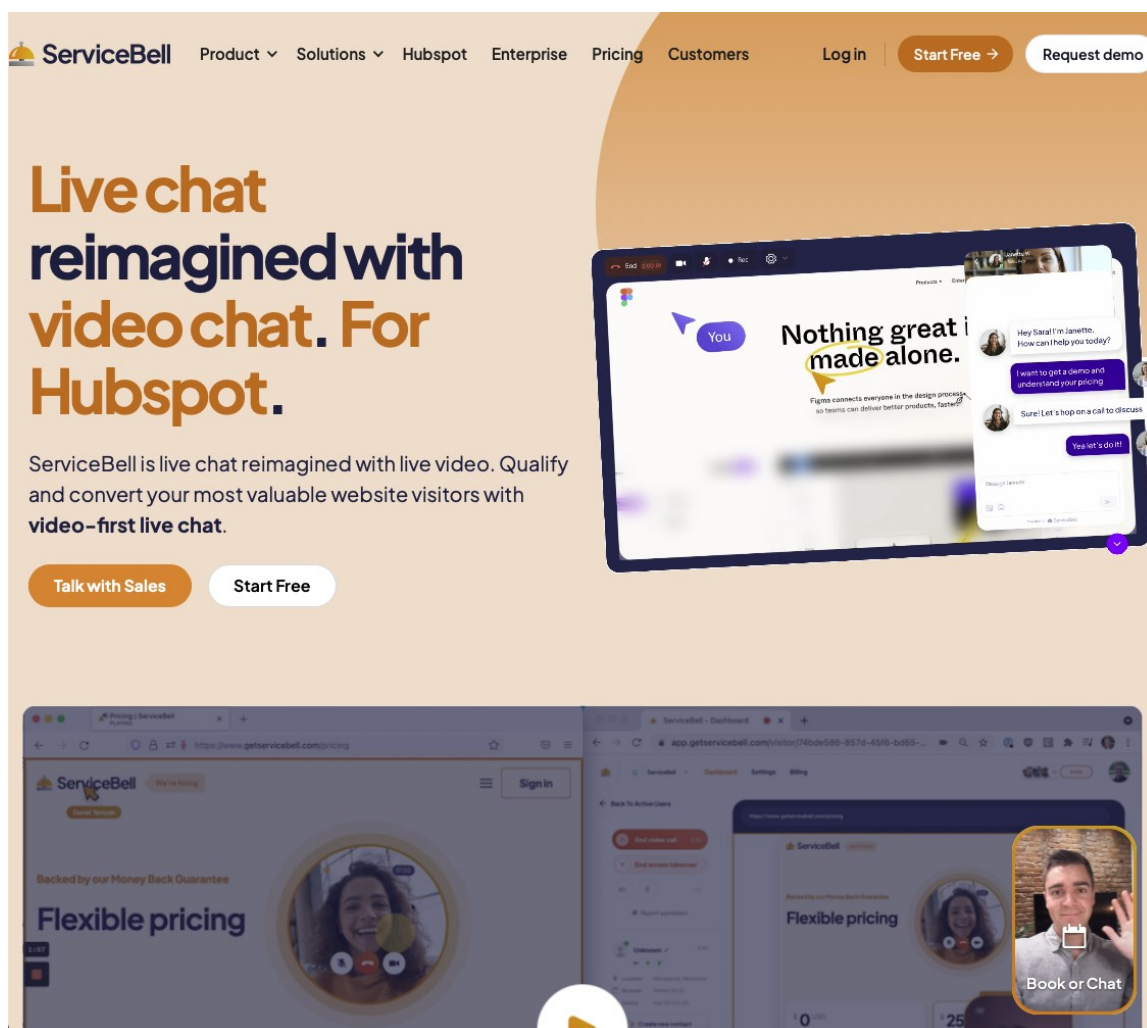
1. VA ei saa olla liian tunkeutuva käyttäjän näkökulmasta.
2. VA:n tulee visuaalisesti sopeutua alkuperäisen ratkaisun ulkonäköön ja suunnitteluideoihin.
3. VA:n tulee herättää huomiota ilman kirkkaita värejä.
4. VA:n täytyy pystyä tasapainottamaan huomautukset ja vinkit niiden vaikutusten perusteella. Vaikutteita voidaan ajatella joko positiivisina, negatiivisina tai neutraaleina.
5. VA:n täytyy pystyä näyttämään viestejä ja tehtäviä.
6. VA:n tulee pystyä erottamaan käyttöliittymän ilmoittamista tiedoista yksi tai enemmän ongelmia, jotka vaativat toimintaa käyttäjältä. Näistä ongelmista avustajan tulee luoda tehtäviä käyttäjälle.
7. VA:n täytyy pystyä asettamaan tärkeät viestit tai tehtävät kiireettömien viestien ja tehtävien edelle.
8. VA on esillä vain näkymissä, jotka ovat henkilökohtaisia. Eli tiimit- ja organisaatio-näkymät eivät sisällä avustajaominaisuutta.
9. VA:n tulee käyttää henkilön yksilölliseen näkymään soveltuvia tietoja.

Työn ensimmäisessä vaiheessa suunnitellaan virtuaalisen avustajan ulkonäkö, toimintaperiaate, määritellään mahdolliset rajoitteet tietojen haussa ja käytössä työhön integroinnissa ja viestintäjärjestelmän toiminta, eli miten ja milloin viestipäivitykset tapahtuvat. Kaikki nämä osiot yhdistettynä tulevat luomaan virtuaalisen avustajan, joka pystyy toimimaan edellä mainittujen tavoitteiden ja rajoitteiden mukaisesti.

## 4.1 Ulkoasu

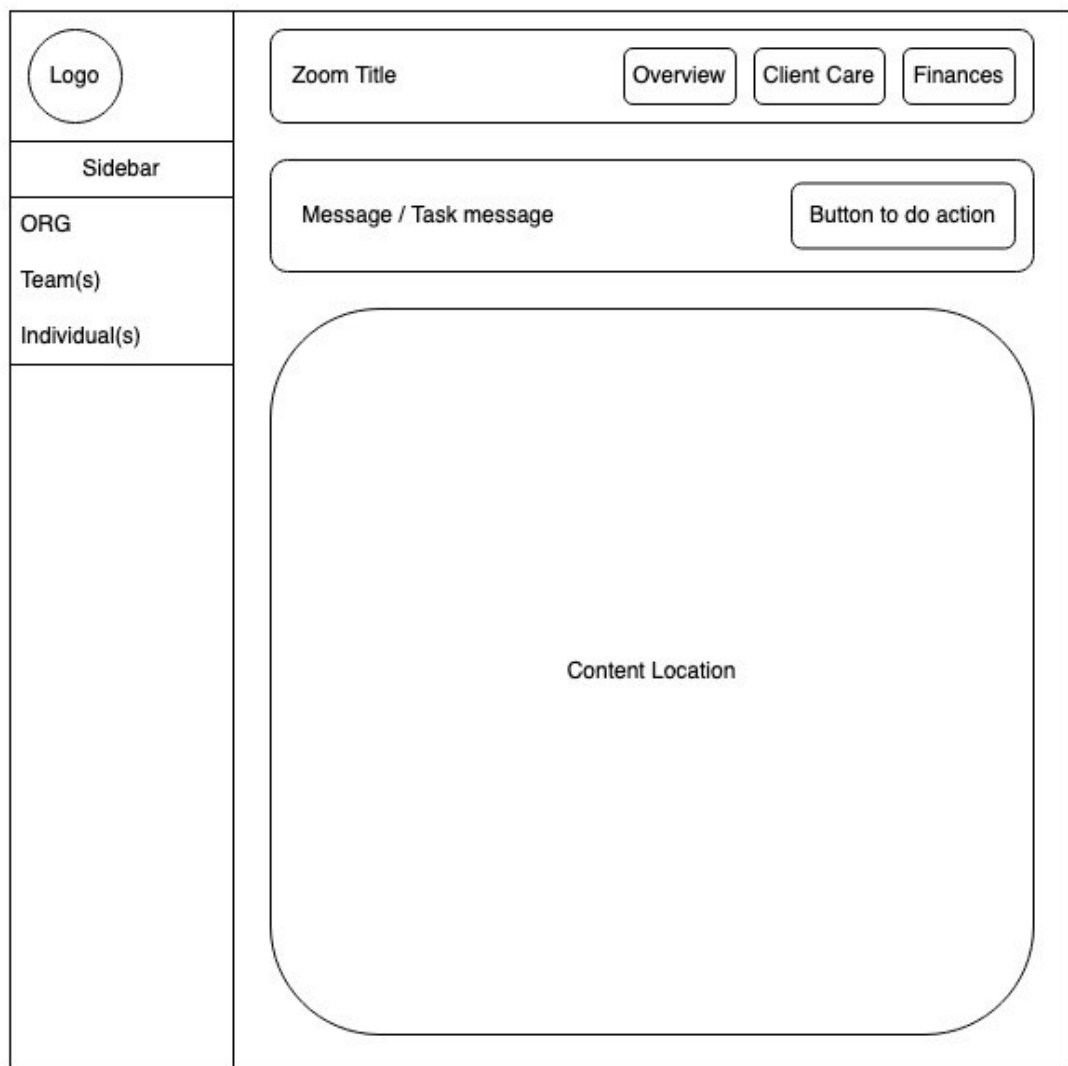
Ulkoasu on avustajan toiseksi suurin osio toiminnallisen osion jälkeen. Kehno ulkoasu voi tehdä muuten hyvästä virtuaalisesta avustajasta keskinkertaisen tai jopa käyttökelvottoman. Syy tähän on se että, virtuaalisen avustajan ulkonäkö vaikuttaa siihen, miten käyttäjä vastaanottaa viestit. Jos VA vie liian paljon huomiota, se voi olla häiritsevää tai jopa ärsyttävää käyttäjälle. Jos taas VA on täysin huomaamaton, käyttäjä voi helposti mennä sen ohi vahingossa. Siksi on tärkeää, että ulkonäköön ja sen tarjoamaan käyttäjäkokemukseen on kiinnitetty runsaasti huomiota.

Ensimmäisenä on päätettävä, minne virtuaalinen assistentti ja assistentin viestit tulee sijoittaa. Monet käyttävät avustajaa keskustelubotin tapaan popup-tyylillä. Tämä vaatisi erillisen napin sijoittamisen ruudun vasempaan tai oikeaan alanurkkaan. Kuva 8 demonstroi, miten ServiceBell käyttää popup-metodia keskustelu- ja ajanvaraustarkoituksiin verkkosivun oikeassa alalaidassa.



Kuva 8. ServiceBell-verkkosivuston keskustelu- ja ajanvarauspainikkeen implementaatio [30]

Kyseinen valinta vaatisi assistentin sijoittamista muiden tietojen päällä, jos se on aktiivisessa tilassa. Tämän lisäksi se tulisi olemaan suuri ongelma käytettäessä avustajaa pienillä päätelaitteilla, kuten älypuhelimilla. Tästä syystä kyseinen vaihtoehto ei ollut mahdollinen valinta toteutukseen. Sen sijaan virtuaalisen assistentin tulisi sijaita työn kategoriapalkissa, joko palkin sisällä tai lisäyksenä eri elementissä. Kuva 9 visualisoi, miten avustaja voitaisiin toteuttaa erillisessä elementissä. Tällöin navigointi ja sivun sisältö pysyisi samana tai samankaltaisena kuin ennen eikä elementti aiheuttaisi toimintojen ja muun sisällön sekoittumista tai häiriintymistä.



Kuva 9. Suunnitelma virtuaalisen avustajan sijoittamisesta erilliseen elementtiin

Itse ulkonäön tulisi pysyä samoissa väreissä kuin muidenkin näkymän osien. Tämä tarkoittaisi sitä, että värimahdollisuudet olisivat valkoinen, harmaa, musta, ja oranssi sekä edellä mainittujen värien eri sävyt. Oranssi väri on liian kirkas ja huomiota herättävä, ja sen käyttö tulisi rajoittaa tiedon

visualisointiin esim. taulukoissa. Näin ollen toimiviksi metodeiksi jää tummempien värisävyvahvistuksien käyttö tai elementtien rajaaminen.

#### 4.2 Teknisten vaatimusten toteutus

Tärkeä toiminnallinen päätös liittyy siihen, miten ja minne avustaja tallentaa tietoa. Tiedon täytyy sisältää seuraavat komponentit:

- Tehtävän tai viestin teksti, jonka avustaja tulee ilmoittamaan
- Tieto viestin tai tehtävän tärkeydestä
- Onko viesti positiivinen, negatiivinen vai neutraali
- Kenelle viesti tai tehtävä on tarkoitettu
- Onko viesti saapunut perille tai onko tehtävä tehty

Taulukko 1 visualisoi, miten tiedon tallentamisen voi hoitaa laskentataulukolla käyttäen edellä mainittuja komponentteja.

Taulukko 1. Tiedon tallennusrakenne laskentataulukkona

Henkilö	Tärkeä	Viestin luonne	Tehty	Viesti
Henkilötieto 1	Kyllä	Positiivinen	Kyllä	Viesti 1
Henkilötieto 2	Ei	Neutraali	Ei	Viesti 2
Henkilötieto 3	Ei	Negatiivinen	Ei	Viesti 3

Edellä mainitut tiedot VA:n täytyy pystyä lukemaan ja ymmärtämään niin, että se voi toimia niiden perusteella. Lisäksi virtuaalisen avustajan tulee pystyä lisäämään käyttäjille sisäisiä tehtäviä tai huomautuksia ilman niiden tallentamista, koska nämä tiedot ovat jo olemassa ja nähtävissä toisissa käyttöliittymän elementeissä. Erittäin hyvä lisäys on VA-ilmoitus käyttäjälle, jos hänellä on

maksamattomia laskuja, tai jos hän on lähettänyt laskuja, joista ei ole vielä saatu maksua tai maksua ei ole merkitty.

Tiedon tallennuksen tulisi tapahtua Google Sheets -palvelussa, koska se on jo valmiina integroituna käyttöliittymän toimintaan. Lähtötilanteessa Google API -oikeudet on rajoitettu vain luku-muotoon. Kirjoitusoikeudet ovat osa Google Sheets API -ominaisuuksia, joten oikeuksia tulee muuttaa sallimaan myös kirjoittaminen. Tämän lisäksi Google Sheets antaa yritykselle helpon ja tutun tavan lisätä tehtäviä ja viestejä muille.

Virtuaalisen avustajan tulisi myös onnitella käyttäjää kaikkien tehtävien suorittamisesta. Tutkimukset tukevat sitä, että työntekijöiden tyytyväisyys kasvaa ja unenlaatu paranevat, jos työntekijä saa kehuja hyvästä työstä [31]. Tästä syystä onnittelutoiminto tulisi integroida osaksi avustajaa. Tämä voi myös parantaa työntekijöiden kuvaa ja mielipidettä virtuaalisesta avustajasta. Tämä toiminto voisi tapahtua yksinkertaisella kehu- tai kiitosviestillä, kun asiat ovat hyvin.

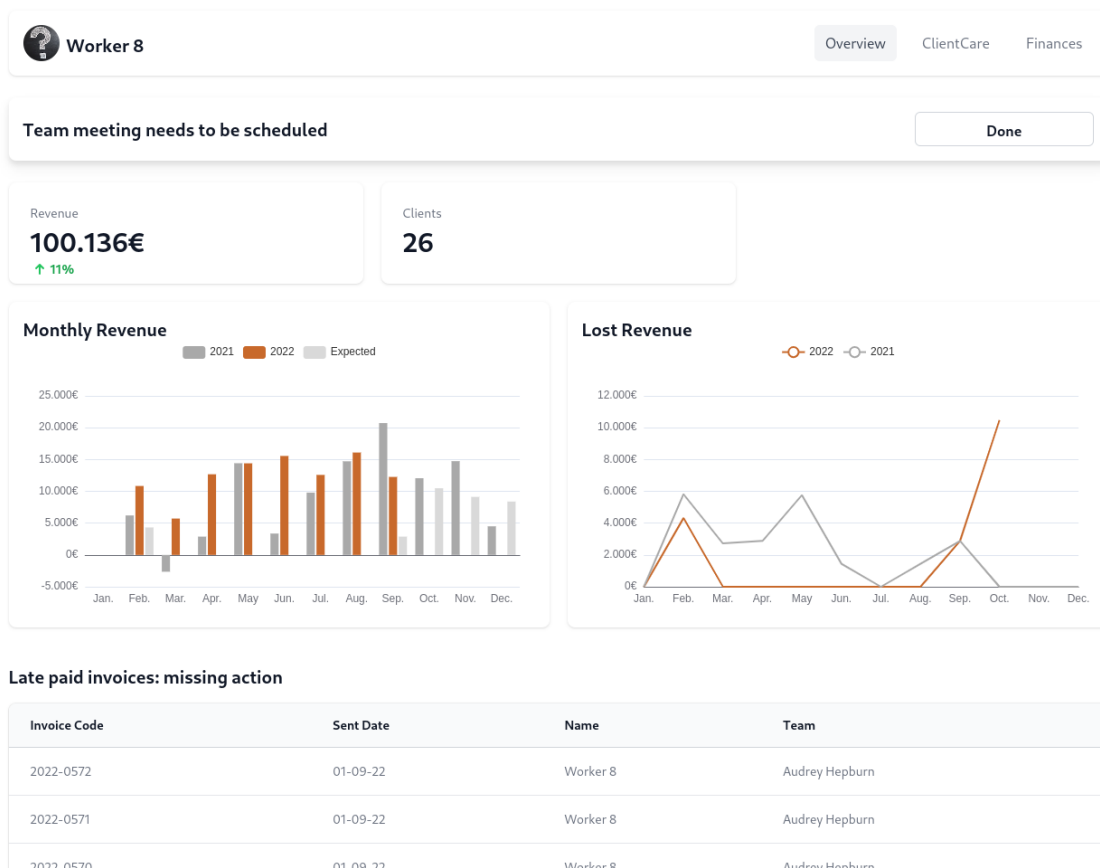
Palvelin edellyttää merkittäviä muutoksia lähtökohtana olevaan ratkaisuun verrattuna. Palvelimen tulee paitsi vastaanottaa tietoa sisältäviä pyyntöjä, myös lähettää pyyntö eteenpäin tai merkitä sitä vastaava viesti tai tehtävä tehdyksi. Kyseinen toiminto vaatii kirjoitusominaisuuden lisäämistä Google Sheets API:n avulla käytössä olevan lukuominaisuuden rinnalle. Tämän lisäksi tarvitaan uusi Google Sheets -sivu ja sen sisältämän tiedon välittäminen palvelimen avulla selainpalvelimelle, kuten muissakin Google Sheets -sivuissa. Loput muutoksista tai ominaisuuksista virtuaalisessa avustajassa voi tehdä selainpalvelimessa. Tämän ei pitäisi vaikuttaa negatiivisesti laitteen toimintakykyyn, ellei tehtäviä lasketa tai suoriteta moneen kertaan muiden laskujen ja toimintojen kanssa.

## 5 Virtuaalisen avustajan toteuttaminen

### 5.1 Ulkoasu

Virtuaalisen avustajan lisääminen alkuperäiseen web-näkymään vaati vain vähän ulkoasuun liittyviä muutoksia. Värisävyjen sijaan työ käyttää Tailwind CSS:n varjo-ominaisuutta. Tällä tavalla virtuaalinen avustaja saadaan erottumaan muista osioista ilman vahvoja värejä. Näin toteutettuna VA-osuus näyttää olevan korkeammalla kuin muut osuudet.

Avustajan viestiteksti on myös samankokoinen kuin muut otsikot (kuva 10). Tämän tarkoitus on hyvin spesifinen: säilyttää viestin herättämä huomio samalla tasolla verrattuna muihin näkymän osioihin. Minkään otsikon tai osion ei tule olla suurempi tai huomiota herättävämpi kuin toisten. Jos osioiden otsikot ovat käyttäjän silmiin miltei samankokoisia, tulee virtuaalisen avustajan osion rajoitus korostamaan tekstin ja koko osion näkyvyyttä. Tällä tavalla eri osiot täydentävät toisiaan eivätkä tarpeettomasti herätä huomiota siinä määrin, että se häiritsisi käyttäjän normaalia toimintaa.

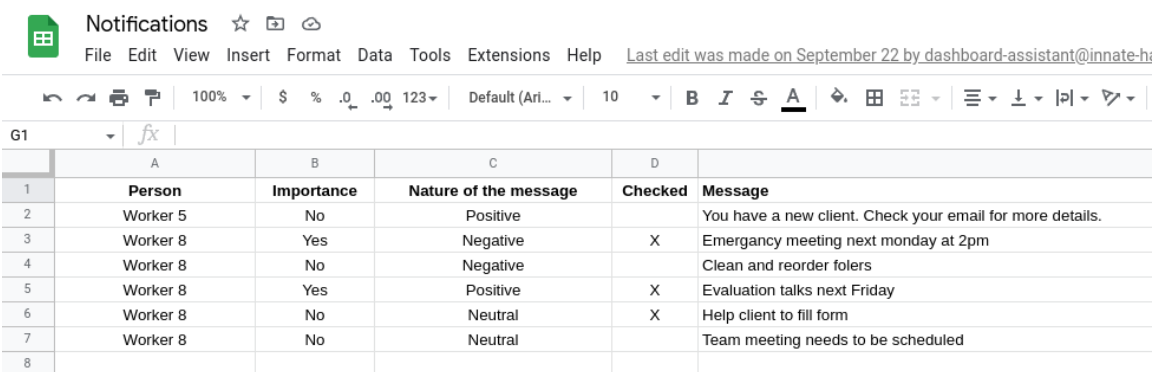


Kuva 10. Virtuaalisen avustajan ulkonäöllinen toteutus

Hyväksyntäpainike ei vaadi paljon huomiota, koska käyttäjä tarvitsee sitä vain silloin, kun tehtävä on suoritettu tai viesti on luettu. Tärkein onkin vain osoittaa, että kyseessä on painike ja että sitä painamalla voi joko kuitata tehtävän suoritetuksi tai siirtyä eteenpäin ehdotuksissa.

## 5.2 Tietokanta

Tietokannan rakenne pysyi lähes muuttumattomana alkuperäiseen suunnitelmaan verrattuna, sillä ainoa tekninen muutos oli kyllä- tai ei-merkinnästä luopuminen tehtävän toteutumisen tai suorittamisen merkitsemisessä. Sen sijaan merkintä tapahtuu yksinkertaisella X-merkillä. Jos tehtävä suoritettu tai viestiin reagoitu, käytetään merkintää X, ja muussa tapauksessa sarake jätetään tyhjäksi (kuva 11).



	A	B	C	D	
1	Person	Importance	Nature of the message	Checked	Message
2	Worker 5	No	Positive		You have a new client. Check your email for more details.
3	Worker 8	Yes	Negative	X	Emergancy meeting next monday at 2pm
4	Worker 8	No	Negative		Clean and reorder folers
5	Worker 8	Yes	Positive	X	Evaluation talks next Friday
6	Worker 8	No	Neutral	X	Help client to fill form
7	Worker 8	No	Neutral		Team meeting needs to be scheduled
8					

Kuva 11. Tietokannan ilmoitustaulukon rakenne

## 5.3 Palvelin

Palvelimelle lisättiin Google Sheets API -oikeudet ja ominaisuudet merkitä tehtävä tehdyksi tai viesti luetuksi sekä lukea viesti- tai tehtävätaulukko.

Oikeuksien päivitys tapahtui muuttamalla OAuth scope spreadsheets -lukumuodosta (read only) spreadsheets-muotoon. Spreadsheets-muoto sallii palvelimen lukea ja kirjoittaa Google Sheets -laskentataulukoihin. Kuva 12 sisältää Oauth-ohjelmakoodin, jota palvelin käyttää identifioidakseen itsensä Googlelle.



```

182 //Google Sheets Auth settings
183 const authentication = async (service) => {
184   var scope;
185   //console.log(service);
186   const auth = new google.auth.GoogleAuth({
187     keyFile: credPath,
188     scopes: "https://www.googleapis.com/auth/spreadsheets"
189   });
190   const client = await auth.getClient();
191   const sheets = google.sheets({
192     version: 'v4',
193     auth: client
194   });
195   //console.log('in sheets');
196   scope = "https://www.googleapis.com/auth/spreadsheets";
197   return {sheets}
198 }

```

Kuva 12. Google Oath -asetukset

Selainpalvelimen kyselyn vastaustoiminto hoituu helposti yksinkertaisella POST-metodilla (kuva 13), joka sisältää muutoskohdan. Tämän jälkeen palvelin hankkii tiedot (kuva 14) Google Sheets -palvelusta, jos niitä ei ole jo valmiiksi saatavilla. Ennen muutos pyynnön lähetystä muutetaan tietoa POST-pyyntöön mukaisesti (kuva 15). Näiden muutoksien jälkeen palvelin lähettää muutoksen Google Sheets -tietoihin ja päivittää omat tietonsa muutoksen jälkeen (kuva 16).

```

625 api.post('/back/notification-update', jsonParser ,function (req, res){
626   if (notificationData === undefined || notificationData.length === 0){
627     refreshNotifications(res, false).then(() => {
628       recieveNotificationUpdate(req.body);
629       refreshNotifications(res, false);
630     });
631   }
632   else{
633     recieveNotificationUpdate(req.body);
634     refreshNotifications(res, false);
635   }
636 });

```

Kuva 13. Selainpalvelimen vastaanottajatoiminto

```

515 async function refreshNotifications(ress, run) {
516   notificationSheetData = await sheetsRefresh(gSheets[0][3], gSheets[1][6], gSheets[2][0]);
517   notificationData = [];
518   arrayNotificationData = Object.values(notificationSheetData);
519   arrayNotificationData = arrayNotificationData[2];
520   for(let counter = 1; counter < arrayNotificationData.length; counter++){
521     if(arrayNotificationData[counter][0] && !arrayNotificationData[counter][3] && arrayNotificationData[counter][4]){
522       notificationData.push({
523         id: counter + 1,
524         person: arrayNotificationData[counter][0],
525         importance: arrayNotificationData[counter][1],
526         nature: arrayNotificationData[counter][2],
527         message: arrayNotificationData[counter][4]
528       });
529     }
530   }
531   notificationData = {notifications: notificationData};
532   if(run === true){
533     ress.end(JSON.stringify(notificationData));
534   }
535 }

```

Kuva 14. Tehtävien ja viestien hakufunktio

```

537 //Notification Mark Function
538 async function recieveNotificationUpdate(receivedData){
539   for(let counter = 1; counter < arrayNotificationData.length; counter++){
540     if(arrayNotificationData[counter][4] === receivedData.message && arrayNotificationData[counter][0] === receivedData.person){
541       await UpdateSheet(gSheets[0][3], gSheets[1][6], 'D'+ parseInt(counter + 1), ['X']);
542     }
543   }
544 }

```

Kuva 15. Funktio tehtävän tai viestin merkintään

```

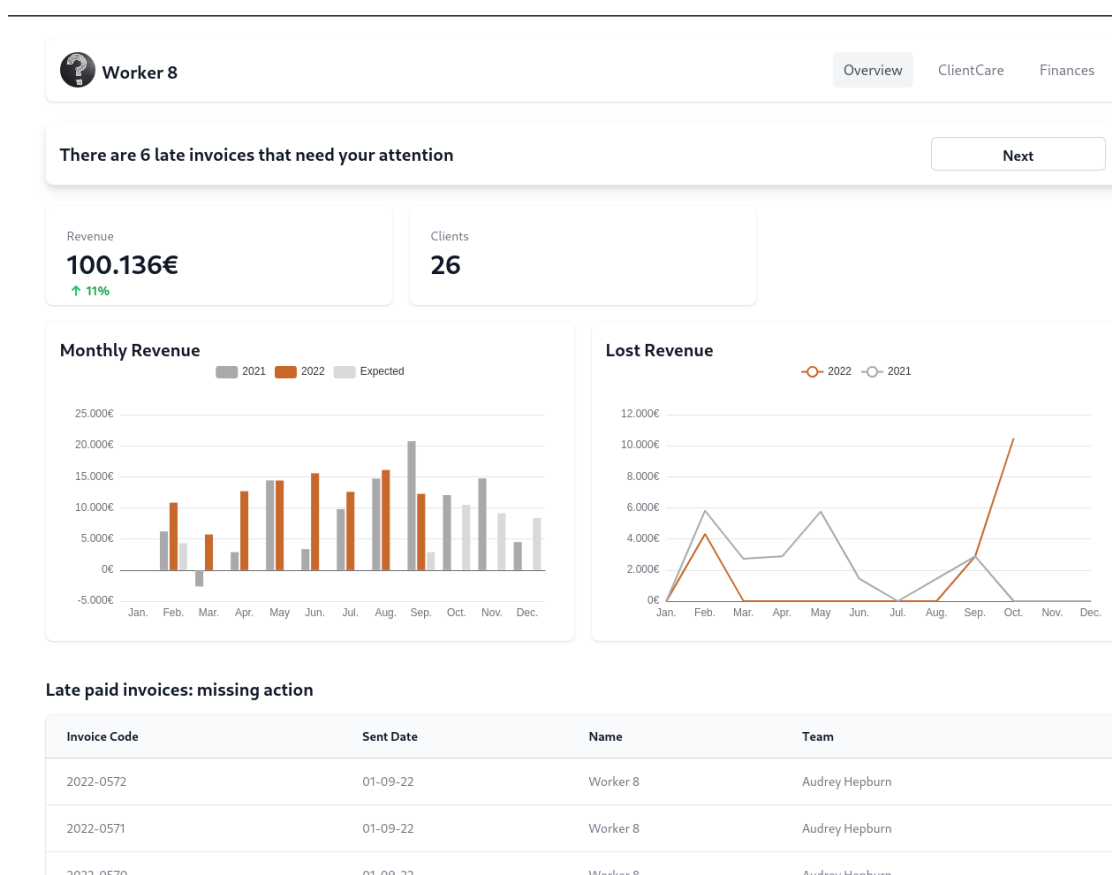
156 //Update Data to Google Sheets
157 async function UpdateSheet(currentID, currentSheet, rangeSpecification, dataValues) {
158   try {
159     const {
160       sheets
161     } = await authentication('sheets');
162
163     const result = await sheets.spreadsheets.values.update({
164       spreadsheetId: currentID,
165       range: currentSheet + '!' + rangeSpecification,
166       valueInputOption: "RAW",
167       resource: {
168         values: [dataValues]
169       }
170     });
171     //console.log('!' + rangeSpecification);
172     //console.log('%d cells updated.', result.data.updatedCells);
173     //Write Command:
174     // await UpdateSheet(gSheets[0][3], gSheets[1][6], '3:3', ['1','Worker 3', 'No', 'Neutral','', 'Test Message',]);
175   } catch (e) {
176     console.log(e);
177     return e;
178   }
179 }

```

Kuva 16. Google Sheets API -päivitystoiminto

## 5.4 Selainpalvelin ja VA-priorisointi

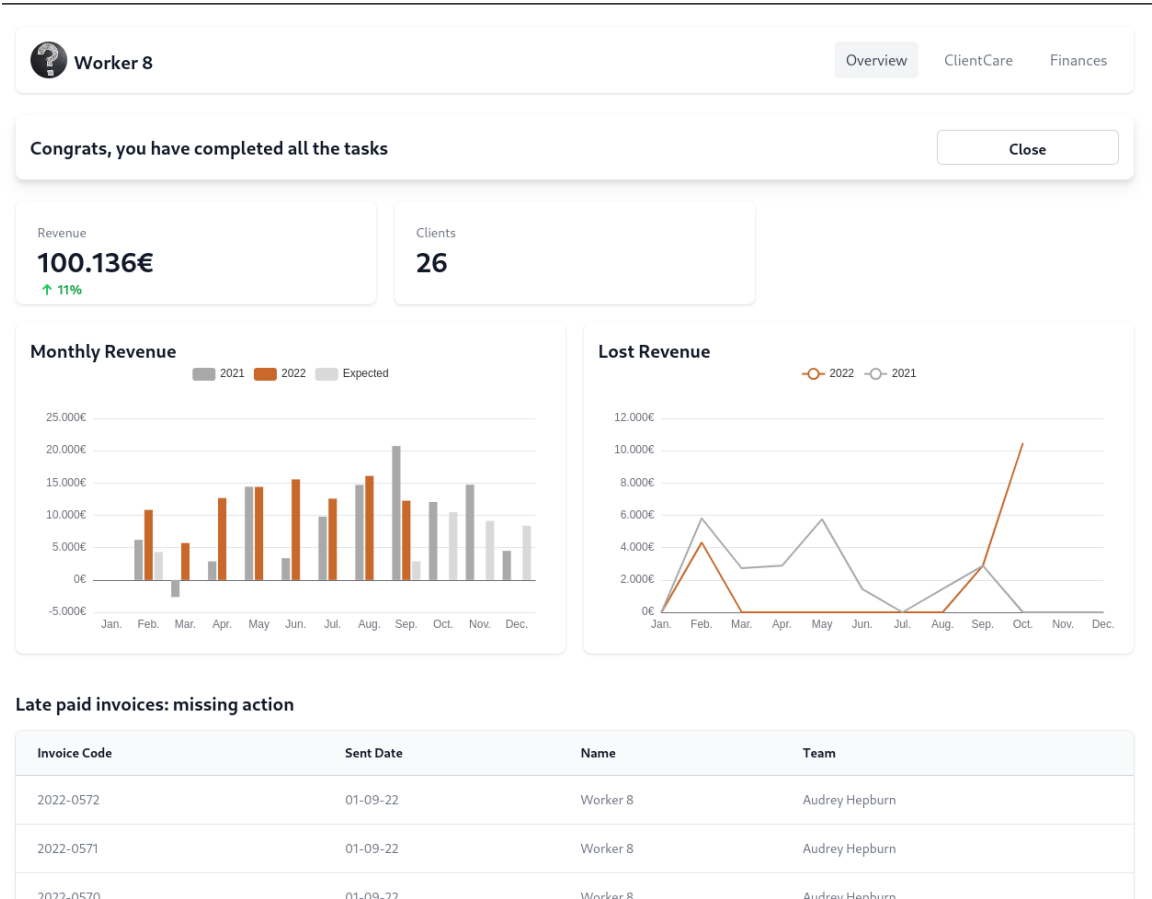
Selainpalvelin ja palvelin yhdessä mahdollistavat tehtävien integroinnin virtuaaliseen avustajaan. Tämä oli yksi tavoitteissa suunnitteluvaiheessa ja toteutettu työn lopputuloksessa. Yksinkertaisuudessaan VA tarkastaa, onko työntekijällä lähetettyjä laskuja, joita ei ole maksettu. Jotta avustaja voi merkitä maksamattomat laskut, se joutuu erottelemaan laskut ehdon avulla: onko laskun lähettämisestä kulunut yli 30 päivää tai onko laskuun merkitty aikaraja ylitetty. Kuva 17 demonstroi, miten avustaja ilmoittaa käyttäjälle maksamattomista laskuista. Kaikki maksamattomat laskut luokitellaan negatiiviseen kategoriaan.



Kuva 17. VA-ilmoitus laskuista, jotka vaativat käyttäjältä huomiota

Laskut päivittyvät automaattisesti palvelimen avulla, koska kaikki tiedot ovat tallennettuna Google Sheets -tietokantaan. Kyseinen toiminto vähentää virtuaalisen avustajan kommunikoinnin määrää Google Sheets -laskentataulukoiden kanssa. Tästä syystä virtuaalinen avustaja voi tuntua hitaalta esimerkiksi tilanteissa, joissa tehtävän merkinnän täytyisi tapahtua alle minuutin sisällä. Tämä näkyy myös tilanteissa, joissa avustaja ei itse kykene tai ei ole suunniteltu tekemään merkintöjä. Esimerkkinä tästä ovat laskut, joiden tilaa (maksettu vai maksamatta) avustaja ei pysty

tarkastamaan pankkitililtä. Käytännössä kaikissa sellaisissa tilanteissa, joissa VA ei pysty merkitsemään muutosta, siirtää Done-painike käyttäjän seuraavaan tehtävään. Jos tehtäviä ei ole enää jäljellä, VA esittää käyttäjälle onnitteluviestin hyvästä työstä (kuva 18). Close-painike sulkee VA-osion kokonaan ja nostaa sivun sisällön sen tilalle. Näin käyttäjä voi nähdä enemmän sivuston sisältöä.



Kuva 18. VA onnittelee käyttäjää hyvästä työstä

Tehtävien ja viestien priorisointi tapahtuu yksinkertaisella luokitteluperiaatteella. Avustaja jakaa tehtävät ja viestit neljään eri kategoriaan: tärkeät, positiiviset, negatiiviset ja neutraalit.

Tärkeäksi kategorioidut viestit ja tehtävät sijoitetaan virtuaalisen avustajan listan kärkeen. Nämä eivät ole riippuvaisia siitä, ovatko ne positiivisia, negatiivisia vai neutraaleja. Sen sijaan järjestys määräytyy merkintäpäivän perusteella: mitä vanhempi tai ylempi tehtävä tai viesti on Google Sheets -listalla, sitä ylempänä se on tärkeyslistalla.

Jos listalla ei ole tärkeäksi kategorioituja tehtäviä tai viestejä, virtuaalinen avustaja alkaa lisäämään positiivisia, negatiivisia ja neutraaleja viestejä sekä tehtäviä vuorotellen. Tällä tavalla VA

kykenee vaikuttamaan käyttäjän mielialaan lievällä tasolla. Järjestelmällä on kuitenkin heikkous. Jos kaikki viestit ja tehtävät ovat negatiivisia, virtuaalinen avustaja ei voi luoda keinotekoisesti positiivisia tai neutraaleja viestejä negatiivisten sekaan. Tämä sama ongelma käy ilmi myös pelkäästään positiivisissa ja neutraaleissa tilanteissa, mutta tämä on harvoin ongelma käyttäjän näkökulmasta.

## 6 Yhteenveto

Työn tavoitteena oli ymmärtää, mikä on virtuaalinen avustaja ja miten sellaisen voisi integroida lähtökohtana olleeseen web-pohjaiseen käyttöliittymiin. Lisäksi tavoitteena oli suunnitella ja toteuttaa yksinkertainen virtuaalinen avustaja täydentämään olemassa olevaa käyttöliittymää, joka soveltuu tiedon suodattamiseen ja visualisointiin. Alkuperäinen ratkaisu ei sisältänyt lainkaan opastusta tai neuvoja käyttäjälle tehtävien suorittamiseen, eikä näkymä myöskään sisältänyt kommunikointimahdollisuuksia missään muodossa.

Työn etenemisprosessi oli hyvin suoraviivainen. Ensimmäisessä vaiheessa selvitettiin, mikä virtuaalinen avustaja (VA) on ja miten se eroaa älykkäästä virtuaalisesta avustajasta (IVA). Suunnitteluvaiheessa rajattiin virtuaalisen avustajan tehtävät sekä määriteltiin toiminnoille asetetut rajoitukset tai vaatimukset. Lisäksi laadittiin suunnitelma avustajan ulkonäöstä, ja Googlen API -rajoi- tuksista saatiin reunaehdot viestintäkonseptille ja säännöt avustajalle. Google API rajoittaa pyyn- töjen määrän 60 sivuun minuutissa, mikä asetti rajat käyttöliittymän päivittämiselle. Säännöt taas olivat ulkoasuun ja toimintaan liittyviä säännöksiä, jotka pohjautuivat pohjatyön tyyli suuntauk- seen. Avustajan täytyy sulautua työn ulkonäköön ja sen ei tule herättää liikaa huomiota väreillä tai muilla visuaalisilla tavoilla tai peittää muita osuuksia työstä. Suunniteltujen toimintojen avulla käyttäjä voi luoda taulukoissa viestin tai tehtävän, jonka avustaja automaattisesti lisää kyseisen käyttäjän tehtäviin tärkeysjärjestyksessä. Yleisesti ottaen virtuaalisen avustajan tulisi herättää huomiota, mutta ei viedä sitä pois muusta työn sisällöstä tai toimia käyttäjää rasittavilla tavoilla, kuten merkkiäänillä tai väri vaihteluilla. Koska työn toteuttamiseen oli käytettävissä rajallisesti ai- kaa, älykkään virtuaalisen avustajan toteutusideasta luovuttiin.

Työn toteutusvaihe onnistui hyvin, minkä taustalla merkittävimminä tekijöinä olivat selkeä suun- nitelma ja toimintakonsepti. Toteutus tapahtui ilman suuria muutoksia suunnitelmaan, ja itse avustajan toiminta täyttää suunnitteluvaiheessa määritellyt tavoitteet ja odotukset. Virtuaalinen avustaja antaa tärkeät tiedot ensin, minkä jälkeen se tarjoaa ne työntekijän näkökulmasta psyko- logisesti positiivisessa järjestyksessä, mikäli se on mahdollista.

Työ on nykyisessä muodossaan toimiva avoimen lähdekoodin pohja virtuaalisen avustajan integ- roimiseksi web-näkymään. Lisäksi työ demonstroi, miten yksinkertainen avustaja pystyy autta- maan uusia sekä kokeneita käyttäjiä. Avustajaa ei ole kuitenkaan rakennettu toimimaan alle mi- nuutin viivevaatimuksen mukaan, joten avustajaa tulee soveltaa tai muokata kohdeprojektin vaa- timuksien mukaan.

## 7 Pohdinta

Työ onnistui ja täytti lähtökriteerit erinomaisesti. Tästä huolimatta työn tuloksena syntynyt virtuaalinen avustaja ei ole täydellinen monessa suhteessa. Jo työn suunnittelun alussa havaittiin, että toteutusvaihtoehtoja on useita. Nämä eri mahdollisuudet jättivät paljon varaa jatkokehittämiselle. Yksi mahdollinen etenemisidea on muuttaa palvelimen kommunikointi websocket-pohjaiseen ratkaisuun. Tämä täyttäisi monet reaaliaikaisuusvaatimukset, jotka voivat olla olennaisia useissa käyttötapauksissa. Suurempi syy on kuitenkin se, että tällä muutoksella avustaja tuntuisi reaktiivisemmalta käyttää ja samalla poistettaisiin tarve päivittää sivua käyttäjän toimesta.

Toinen idea on virtuaalisen avustajan muuntaminen älykkääksi virtuaaliseksi avustajaksi. Tekoälyn lisääminen avustajaan olisi luonnollinen muutos. AI:n integrointi työhön voisi parantaa käyttäjäkokemusta huomattavasti. Tähän tarkoitukseen tekoälyn ei tarvitse olla äärimäisen laaja, vaan sen sijaan tiettyihin kohteisiin tai aiheisiin fokusoitu versio sopisi erinomaisesti. Fokusoinnin alue voisi esimerkiksi olla työntekijöiden ohjaus tai laskujen ja maksujen käsittely.

Käyttöliittymän ja avustajan personalisointi tarjoaa myös kehittämismahdollisuuksia. Personalisointiominaisuus vaatisi oikeuksien luomisen esimerkiksi Google-tilien avulla. Työn lopputuloksena syntyneessä ratkaisussa kuka vain voi katsoa toisen henkilön tietoja ja avustajan viestejä. Tietojen avoimuus voi luoda yrityksessä ongelmia käyttäjille, jotka haluavat pitää tietonsa yksityisinä. Jos käyttäjän katseluoikeuksia kyettäisiin rajoittamaan, näkisi käyttäjä tapauskohtaisesti omien tietojensa lisäksi myös tiimin ja yrityksen tiedot. Tällä muutoksella voitaisiin luoda rennompi työympäristö, sillä se saattaisi vähentää käyttäjien vertailua toisiinsa, ja siten nostaa työtyytyväisyyttä. Tämä pätee myös virtuaaliseen avustajaan: neuvojen tulee olla näkyvissä vain kirjautuneille henkilöille tietyissä rooleissa. Nämä voisivat olla esimerkiksi organisaation johtajat ja tiimin johtajat.

Google-tilillä kirjautumisen lisäys mahdollistaa myös pilvisynkronoinnin. Tällä ominaisuudella annettaisiin käyttäjille mahdollisuus ansaita suorituspalkintoja tai määritellä tilille henkilökohtaiset asetukset, jotka tallentuisivat pilveen. Näin jokainen voisi räätälöidä avustajan toimivuutta henkilökohtaisiin tarkoituksiin ja asetukset pysyisivät laiteelta toiselle automaattisesti. Lisäksi virtuaalinen avustaja kykenisi muistamaan, mitä käyttäjälle on jo aiemmin ilmoitettu. Näin avustajasta tulisi älykkäämpi ja mukavampi työkaveri, koska se ei toista ohitettuja asioita uudestaan.

Yllä on esitetty vain muutamia ideoita yksinkertaisen virtuaalisen avustajan jatkokehitykselle. Mitä nopeammin AI-teknologia kehittyy ja yleistyy, sitä enemmän uusia kehitys- ja sovellusmahdollisuuksia AI-teknologialle avautuu myös virtuaalisten avustajien yhteydessä.



## Lähteet

- (1) Tietotekniikan sanasto: hakusana Intelligent Virtual Assistant (IVA). TTEC's glossary. TTEC 2023. Saatavilla: <https://www.ttec.com/glossary/intelligent-virtual-assistant> [viitattu 18.12.2022].
- (2) What is artificial intelligence and how is it used?. 2021. European Parliament. Uutisotsikko. Saatavilla: <https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used> [viitattu 22.4.2023].
- (3) What is a Digital Assistant? 2023. Workgrid. Saatavilla: <https://www.workgrid.com/article/what-is-a-digital-assistant> [viitattu 22.4.2023].
- (4) Tietotekniikan sanasto: hakusana API. Cambridge dictionary. Cambridge 2023. Saatavilla: <https://dictionary.cambridge.org/dictionary/english/api> [viitattu 22.4.2023].
- (5) Tietotekniikan sanasto: hakusana http. Cambridge dictionary. Cambridge 2023. Saatavilla: <https://dictionary.cambridge.org/dictionary/english/http> [viitattu 22.4.2023].
- (6) Tietotekniikan sanasto: hakusana Runtime Environment. Techopedia dictionary. Techopedia 2023. Saatavilla: <https://www.techopedia.com/definition/5466/runtime-environment-rte> [viitattu 22.4.2023].
- (7) Introducing JSON. The JSON data interchange syntax. 2nd edition. 2017. Saatavilla <https://www.json.org/json-en.html> [viitattu 22.4.2023].
- (8) Tietotekniikan sanasto: hakusana CSS. Cambridge dictionary. Cambridge 2023. Saatavilla: <https://dictionary.cambridge.org/dictionary/english/css> [viitattu 22.4.2023]
- (9) Siri. 2023. Apple. Saatavilla: <https://www.apple.com/siri/> [viitattu 29.4.2023].
- (10) ChatGPT Could Be AI's iPhone Moment. 2022. Bloomberg. Uutisotsikko. Saatavilla: <https://www.bloomberg.com/news/newsletters/2022-12-12/chatgpt-the-gpt-3-chatbot-from-openai-microsoft-is-tech-magic> [viitattu 18.12.2022].

- (11) IBM Shoebox. 2023. IBM. Saatavilla: [https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1\\_7.html](https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html) [viitattu 29.4.2023].
- (12) Amazon Echo. 2023. Amazon. Saatavilla: <https://www.amazon.com/dp/B07XKF75B8> [viitattu 29.4.2023].
- (13) Intelligent Virtual Assistant Market Size, Share & Trend Analysis Report By Technology (Text-to-Speech, Text-based), By Product (Chatbot, Smart Speaker), By Application (IT & Telecom, Consumer Electronics), And Segment Forecasts, 2023 – 2030. 2023. Grand View Research. Raportti. Saatavilla: <https://www.grandviewresearch.com/industry-analysis/intelligent-virtual-assistant-industry> [viitattu 28.4.2023]
- (14) The Rapid Rise of Virtual Assistants in Different Industries. Saatavilla: <https://www.eleviant.com/insights/blog/the-rapid-rise-of-virtual-assistants-in-different-industries/> [viitattu 8.12.2022].
- (15) The Bots Are Coming (to Assist). Saatavilla: <https://www.ttec.com/blog/bots-are-coming-assist> [viitattu 18.12.2022].
- (16) How Kindly Manages 90% Of NBX's Inbound Customer Support Enquiries. 2023. Kindly. Tapaustutkimus. Saatavilla: <https://www.kindly.ai/case-studies/chatbot-norwegian-block-exchange> [viitattu 29.4.2023].
- (17) Virtual Assistants and their Use-cases in different industries. 2022. Geakminds. Blogi. Saatavilla: <https://geakminds.com/use-cases-of-virtual-assistants-in-various-industries/> [viitattu 18.12.2022].
- (18) Technical support services. 2023. Nokia. Saatavilla: <https://www.nokia.com/networks/mobile-networks/technical-support-services/> [viitattu 29.4.2023].
- (19) Full Stack Development Explained. 2023. MongoDB. Saatavilla: <https://www.mongodb.com/languages/full-stack-development> [viitattu 29.4.2023].
- (20) What Is a Database?. 2023. Oracle. Saatavilla: <https://www.oracle.com/database/what-is-database/> [viitattu 29.4.2023].
- (21) Google Sheets API Overview. 2022. Google. Saatavilla: <https://developers.google.com/sheets/api> [viitattu 18.12.2022].

- (22) Service Accounts. 2022. Google. Saatavilla: <https://developers.google.com/shopping-content/guides/how-tos/service-accounts> [viitattu 18.12.2022].
- (23) About Node.js®. 2022. NodeJS. Saatavilla: <https://nodejs.org/en/about/> [viitattu 18.12.2022].
- (24) Usage limits. 2022. Google. Saatavilla: <https://developers.google.com/sheets/api/limits> [viitattu 18.12.2022].
- (25) What is React.js? (Uses, Examples, & More). Saatavilla: <https://blog.hubspot.com/website/react-js> [viitattu 18.12.2022].
- (26) JavaScript History. 2022. W3 School. Saatavilla: [https://www.w3schools.com/js/js\\_history.asp](https://www.w3schools.com/js/js_history.asp) [viitattu 18.12.2022].
- (27) Code Institute What is javaScript? Saatavilla: <https://codeinstitute.net/nl/blog/what-is-javascript-and-why-should-i-learn-it/> [viitattu 18.12.2022].
- (28) Tailwind CSS: What It Is, Why Use It & Examples. Saatavilla: <https://blog.hubspot.com/website/what-is-tailwind-css> [viitattu 18.12.2022].
- (29) How Does Red Color in Web Design Attract Your Visitors. Saatavilla: <https://www.freepik.com/blog/red-color-web-design-attract-visitors/> [viitattu 18.12.2022].
- (30) Live chat reimaged with video chat. 2023. ServiceBell. Saatavilla: <https://www.servicebell.com> [viitattu 29.4.2023].
- (31) Is Saying “Thank You” the Secret to Improving Employee Health?. 2019. Safestart. Blogi. Saatavilla: <https://safestart.com/news/is-saying-thank-you-the-secret-to-improving-employee-health/> [viitattu 29.4.2023].

## 8 Liitteet

Opinnäytetyön lähdekoodi: <https://github.com/RaneHyv/dashboard>

Käytetyt taulukot:

- Notifications: <https://docs.google.com/spreadsheets/d/1AX-bUFA5C7zeg2SLKtlwktbu69yNpyoFTAhArgXZzVec/edit#gid=0>
- Clients: <https://docs.google.com/spreadsheets/d/1Mj5QtLdKBpbhM-LxFnFtQok2lzoR0lyPII1lzUIPyh8w/edit#gid=0>
- Finances: <https://docs.google.com/spreadsheets/d/1q0xQlpadb9RfUD011MUSW7h2eXwyuXk5E1k09goTPeQ/edit#gid=0>
- Organization: <https://docs.google.com/spreadsheets/d/17JXwCISkWB-49HYcH1GrYF782Y9toAh8NJPoQzAzwJE/edit#gid=0>