

Kohdataan Some

Tekninen määrittely

Kohdataan Some

Tekninen määrittely

Tiedostot:

- Tekninen määrittely (tämä tiedosto)
- Käyttötapauskaaviot
- Toiminnallisuudet ja huomiot -listaus
- InVision App -prototyyppi (<http://kohdataan-proto.perfekt.io>)

Sisällysluettelo

Johdanto

- Taustaa
- Teknisen määrittelyn dokumentin tarkoitus
- Tiivistelmä
- Yhteiskehittäminen
- Teknisen toteutuksen riskitekijät

Saavutettavuus

- Kilpailutuksen ehtojen vaatimukset
- Saavutettavan toteutuksen tekniset haasteet

Tietosuoja, käyttöehdot ja rekisteriseloste

Tekninen toteutus ja arkkitehtuuri

- Mikropalveluarkkitehtuuri ja kontittaminen
- Rekisteröityminen
- Chat-toiminnallisuus
- Bottitoiminnallisuudet
- Muita keskeisiä toiminnallisuuksia

Johdanto

Taustaa

Tämän dokumentaation tarkoituksena on kuvata Kohdataan-hankkeessa toteutettavaa saavutettavaa somepalvelua (<https://kohdataan.fi/mita-teemme/>) sen tekniselle toteuttajalle mahdollisimman kokonaisvaltaisesti.

Palvelun päämäärä on tukea vuorovaikutusta ja uusia kohtaamisia. Tyytyväisten käyttäjien ja uusien kohtaamisten lisäksi palvelun onnistumista määrittää kyky tehdä palvelu tasavertaisesti saavutettavaksi erityyppisille käyttäjäryhmille. Palvelua on suunniteltu alusta asti yhdessä käyttäjien kanssa ja yhteiskehittäminen eri käyttäjä- ja yhteistyötoimijoiden kanssa jatkuu myös tiiviisti varsinaisen teknisen kehitysvaiheen aikana.

Teknisen määrittelyn dokumentin tarkoitus

Tämä dokumentaatio toimii tukena hankkeen teknisen toteutuksen kilpailutukselle, pohjana tulevien ketterän kehityksen vaiheiden suunnittelulle, työkaluna hankekumppaneiden vuoropuhelun syventämiselle teknisen kehitystyön edetessä sekä liitteineen koontina jo tehdystä suunnittelu- ja käyttäjähaastattelutyöstä.

Tämä dokumentaatio on tarkoitettu luettavaksi yhdessä mukana toimitettavien käyttötapauskaavioiden, toiminnallisen prototyypin kanssa (<http://kohdataan-proto.perfekt.io>) ja toiminnallisuuslistauksen kanssa. Edellä mainittuihin liitteisiin on syytä tutustua ensimmäistä kertaa jo ennen tämän dokumentin lukemista. Toiminnallisen määrittelyn yhteyteen on kuvattu joukko yksityiskohtaisiakin määrittelyjä käyttäjärajapinnoista. Nämä dokumentit ovat tähänastisen suunnittelutyön, käyttäjähaastatteluiden sekä käyttäjätestauksen tulosta ja niiden on tarkoitus toimia pohjana alkuvaiheen työskentelylle. Tuleva työskentely toteutetaan kuitenkin ketterän kehityksen perusteita noudattaen ja täten määrittely voi näiltäkin osin muuttua hankkeen edetessä.

Hankkeen toteutustavasta ja tunnettujen resurssien rajallisuudesta johtuen saatetaan joissain kohdin kuvailla yksityiskohtaisemmin teknologiavalintoja, joiden avulla voidaan mahdollisesti tehostaa ohjelmistokehitystä käyttäen valmiita ohjelmistorajapintoja, -kehyskiä ja -kirjastoja. Nämä esitetyt teknologiavalinnat eivät kuitenkaan rajoita lopullisia toteutettavaan järjestelmään tehtäviä valintoja.

Tiivistelmä

Palvelu toteutetaan "mobile first" web-sovelluksena loppukäyttäjälle. Palvelun ylläpidon roolit toteutetaan todennäköisimmin pääsääntöisesti työpöytäkäyttöön tarkoitettuna

web-sovelluksena. Rekisteröityminen on eritelty tässä dokumentaatiossa erityisesti mahdollisen kolmannen osapuolen SSO-palvelun käytön harkinnan vuoksi. Jos rekisteröityminen päädytään toteuttamaan suoraan järjestelmän omana palveluna yhdistyy rekisteröityminen teknisen toteutuksen osalta kiinteämmin osaksi profiilin luomista.

Yhteiskehittäminen

Palvelun tekninen toteuttaja sitoutuu pitämään toteutettavan palvelun ajantasaisen lähdekoodin sekä tiedossa olevan määrittelyn mukaisen backlogin julkisesti saatavilla koko kehityksen ajan. Palvelussa käytettävistä avoimen lähdekoodin lisensseistä sovitaan erikseen hankkeesta vastaavien toimijoiden kanssa ennen kehitystyön alkamista. Mahdollisesti palvelussa käytettävien kolmannen osapuolen avoimen lähdekoodin ohjelmistojen ja kirjastojen osalta palvelun tekninen toteuttaja sitoutuu selvittämään näiden yhteensopivuuden hankkeen tarpeisiin. *Aidon yhteiskehittämisen hengen mukaisesti teknisen toteuttajan tulee myös osaltaan edesauttaa aktiivisen kehittäjäyhteisön mahdollisuutta osallistua hankkeen tekniseen toteutukseen. Rooleista hankekumppaneiden vuorovaikutuksessa kehittäjäyhteisön kanssa on myöskin hyvä sopia ennen teknisen kehityksen aloittamista.*

Teknisen toteutuksen riskitekijät

Hankkeen suunnitteluvaiheessa esille noussut suurin yksittäinen riskitekijä teknisen toteutuksen kehitysvaiheelle on hankkeen ehdoton resurssien rajallisuus. *Tekninen toteutus pitää täten priorisoida siten, että myös saavutettavuuden ehdot ja tavoitteet toteutuvat aina kun toteutetaan yhtenäinen kokonaisuus toiminnallisuuksia.* Saavutettavuus ja palvelun eri käyttötavat tulevat arviolta viemään suurehkon osan käytettävissä olevista kehitysresursseista.

Saavutettavuus

Kilpailutuksen ehtojen vaatimukset

Palvelu toteutetaan kokonaisuudessaan noudattaen WCAG2.1 standardin (<https://www.w3.org/TR/WCAG21/>) mukaista ohjeistusta. Edellä mainitun ohjeistuksen lisäksi tulee saavutettavuuden osalta huomioida eri käyttäjäryhmien mahdollisimman tasavertainen kyky kommunikoida palvelua käyttäen. Jokaisen kehityssprintin suunnittelussa tulee huomioida toteutettavat toiminnallisuudet myös saavutettavuuden näkökulmasta. Näistä joitain keskeisimpiä on tarkennettu myös tämän dokumentin liitteenä olevassa toiminnallisuuslistauksessa.

Saavutettavan toteutuksen tekniset haasteet

Siinä missä edellä mainittu saavutettavuuden ohjeistus asettaa joukon vaatimuksia järjestelmän tekniselle toteutukselle, tulee teknisen toteutuksen kumppanin huomioida myös muut web-sovelluksena toteuttamisen tekniset haasteet heti alusta alkaen. Monet järjestelmään jo suunnitellut toiminnallisuudet, kuten eri viestintätavat, ovat riippuvaisia käyttäjän päätelaitteeseen pyydettävistä oikeuksista (kuten kamera ja mikrofoni). Niiden tuominen osaksi suunnittelua ja siten käyttöönoton yleistä flowta vaikuttaa merkittävästi käyttäjäkokemukseen. Palvelua on kyettävä käyttämään hyvin laajasti eri laitteilla (eri Android ja iOS versiot sekä eri selaimet), tämän asettamien vaatimuksien vaikutuksista kehitysresursseihin tulee käydä keskustelu hankeosapuolten kesken ennen teknisen kehitystyön alkamista.

Tietosuoja, käyttöehdot ja rekisteriseloste

Palvelun tekninen toteuttaja sitoutuu luomaan palveluun lakisääteiset käyttöehdot, tietosuojaselosteen ja mahdolliset muut palveluun vaadittavat lakisääteiset dokumentaatiot. Kohdataan hankkeessa toimivat vastuuhenkilöt sitoutuvat selvittämään ja toimittamaan näihin dokumentteihin vaadittavat tiedot, jotka liittyvät muiden hankkijapuolen toimijoiden kautta tuleviin reunaehtoihin, kuten esimerkiksi tiedon ja palvelimen sijaintiin sekä tietosuojalausekkeen mukaisiin vaatimuksiin rekisterinpitäjistä ja henkilötietojen käsitteijästä. Järjestelmässä käsitellään ja osittain tallennetaan käyttäjien yksityisiä henkilökohtaisia tietoja viestin ja profiilitietojen muodossa.

Järjestelmän toteutuksessa on pidettävä tietosuoja keskeisenä tekijänä ja jopa ylitettävä GDPR-ehtojen asettamat vaatimukset. Kun tietosuojaan panostetaan, järjestelmä herättää käyttäjissään luottamusta ja on ehdottoman turvallinen paikka yksityisellekin kommunikaatiolle.

Tällaiset lakien asettamat vaatimukset ylittävät tekniset ominaisuudet voivat liittyä esimerkiksi siihen miten parhaiten toteutetaan yksityisten viestien suojele julkisia tai osittain julkisia päätelaitteita käyttäessä, tai siihen miten estetään tilien luvaton käyttö myös kirjautumistietojen joutuessa väärin käsiin.

Tekninen toteutus ja arkkitehtuuri

Mikropalveluarkkitehtuuri ja kontittaminen

Alkuvaiheessa palvelua tullaan ajamaan Kehitysvammaliiton Papunet-palvelimilla, mutta kehityksessä tulee ottaa huomioon myös mahdollisuus siirtää palvelu helposti eri alustalle. Siirrettävyyden ja myöhemmin tässä dokumentaatiossa esitettyjen taustatoiminnallisuuden jakautumisesta johtuen (esimerkiksi erillinen chat-palvelinohjelmisto) on näiden erillisten backend-palveluiden pystyttäminen kontteihin, eli esimerkiksi Docker-virtualisointialustan containereihin, luonteva arkkitehtuuriratkaisu.

Hyvää

- + Modulaarisuus helpottaa muokattavuutta jatkossa, esimerkiksi chat-alusta voidaan vaihtaa
- + Skaalautuvuus
- + Vähennetään konfiguroinnin ja asennusprosessien määrää.
- + Riippumattomuus ympäristöstä
- + Palvelut ovat helposti siirrettävissä uuteen ympäristöön, kuten pilvipalveluun.

Huonoa

- Ylläpidon tietoturvariskien hallintaan kiinnitettävä mahdollisesti uudella tapaa

Rekisteröityminen

Tässä kuvattu rekisteröityminen liittyy järjestelmää varten luotavien kirjautumistunnuksien teknisten toteutustapojen vertailuun. Käyttäjien tunnistaminen on tärkeää osan suunniteltujen ominaisuuksien tarpeisiin, kuten turvallinen tapahtumien järjestäminen. Osa profiiliin tarvittavista tiedoista voidaan saada suoraan rekisteröitymisen kautta käyttäen valikoidusti kolmannen osapuolen SSO-palvelua. Tämä osio ei käsittele teknisiä ratkaisuja, jotka liittyvät ainoastaan profiilin hallintaan. Kolmannen osapuolen SSO-palvelun käyttö mahdollistaa myös perinteisen käyttäjänhallinnan tarpeet, kuten salasanan palauttamisen sähköpostin avulla ja mahdollisesti myös palvelun ikä- ja paikkakuntaperusteisen yksilöinnin.

Erilaiset rekisteröitymiseen kuvatut toteuttamistavat on tuotu esille niiden käyttäjäkokemukseen liittyvän vaikutuksen sekä teknisen toteutuksen eroavien resurssitarpeiden vuoksi.

Social-login

Facebook, Google, auth0

Hyvää

- + Helpottaa rekisteröitymisen toteuttamista
- + Käyttäjän ei tarvitse muistaa tai luoda uusia tunnuksia
- + Palvelun käyttöönotto sujuvampi käyttäjälle
- + Palveluissa yleisesti hyvä perus tietoturva
- + Käyttäjänkonversio on parempi kun tunnukset ovat monilla jo valmiina

Huonoa

- Vaikutukset yksityisyydensuojaan
- Käyttäjien vaikutelma turvallisuudesta ja yksityisyydestä palvelussa

Yle-tunnus (tai muu luotettava ja neutraali toimija)

Hyvää

- + Enimmäkseen samat hyödyt kuin Social-logineissa
- + Kotimaisen toimijan näkyvyys
- + Neutraali ja luotettava toimija

Huonoa

- Jos palvelun käyttöaste on pieni niin käyttäjien näkökulmasta kirjautuminen on yhtä työläs
- Ei laajaa käyttäjäkuntaa, joten suurimman osan käyttäjistä pitäisi kuitenkin luoda tunnukset erikseen

Itse rakennettu käyttäjähallinta (web-frameworkit)

Rails/Devise, Node/Passport, Django

Hyvää

- + Käyttäjien yksityisyydensuoja hallitaan täysin
- + On mahdollista tehdä myös valinta, jossa käyttäjähallinta tehdään aluksi paikallisesti ja palvelu tarjoaa rajapinnan tulevaa Social-login -integraatiota varten (esim. Passport)

Huonoa

- Vaatii huomattavasti enemmän työtä rekisteröitymisen ja käyttäjähallinnan osalta
- Perustason tietoturvan rakentaminen on työläämpää
- Profiilia varten saatavia tietoja ei tule valmiina

Hybridiratkaisut (SSO sekä oma rekisteröinti)

Hyvää

- + Jätetään valinta helppouden, yksityisyydensuojan ja käytön vaivattomuuden osalta käyttäjälle
- + Käyttäjäkonversio

Huonoa

- samat huonot puolet kuin social logineissä

Istunnon hallinta

- Esim. JSON Web Tokens
 - Käyttäjien toimiessa selainpohjaisessa web-applikaatiossa tulee istuntojen hallinnan vaikutuksiin kiinnittää huomiota kehitysvaiheessa
 - Onko käyttäjä esimerkiksi yhteiskäyttöpäätteellä vai omalla mobiililaitteellaan

Chat-toiminnallisuus

Palvelun tarpeisiin vaadittavat chat-toiminnallisuudet ovat hyvin perustasoisia useimmissa viestiohjelmassa (esim. Whatsapp tai Telegram) ilmeneviä ominaisuuksia. Tämänkaltaisia ominaisuuksia ovat esimerkiksi viestien lähettäminen ryhmissä ja kahdenkeskisesti sekä erilaiset viestimuodot, kuten teksti-, ääni- ja kuvaviestit. Täten palvelun toteutuksessa voidaan saavuttaa merkittäviä tehokkuushyötyjä valitsemalla taustalle jokin valmiit rajapinnat toteuttavat chat-framework. Kuitenkin ennen valitun frameworkin käyttöönottoa tulee vielä kertaalleen varmistaa ettei tiedossa olevat mahdolliset jatkokehitystarpeet esty valitun frameworkin vuoksi ja myös ohjelmiston rakenteessa huomioida mahdollisuus vaihtaa frameworkkiä jos tällainen tilanne tulisi myöhemmin eteen. Frameworkin valinnassa tulee ottaa huomioon myös järjestelmän erityistarpeet bottien ja ylläpidon osalta (esim. ylläpidon/botin kutsuminen ryhmään).

Palvelinpuolen chat-toiminnallisuus

Valmiin open source chat-palvelimen käyttö

Tämä vaihtoehto on esitetty erityisesti helpoimman käyttöönoton vuoksi. Monet ainoastaan räätälöityjä chat-palveluita tuottavat kaupalliset toimijat ovat avanneet tarjoamissaan palveluissa käyttämänsä ohjelmistot avoimeksi ja ne ovat optimoitu juuri räätälöityjen chat-palveluiden luomiseen. Kehitysresurssien säästämisen kannalta tämäntyyppinen valinta olisi todennäköisesti paras.

- Rocket.Chat, Zulip, Mattermost, Let's Chat, Matrix, Skygear

Hyvää

- + iso osa tarvittavista kanava- ja viestimistoiminnallisuuksista saadaan valmiina
- + kypsässä kehitysvaiheessa olevan palvelun tietoturva on yleensä hyvä ja toiminnallisuudet vakaita

Huonoa

- palvelun toiminnallisuuksissa voi olla ylimääräistä joka joudutaan estämään erikseen
- onko kaikki toteutukseen tarvittava rajapinnoissa?

Palveluntarjoajan pilvessä hostaama chat-alusta

Tämä vaihtoehto vaatisi käytännössä kumppanuuden sopimisesta palveluiden yleisen lisenssi-perusteisen hinnoittelun vuoksi (esim. Sendbird, Pusher tai Ninchat).

Hyvää

- + Hyvät puolet kuten yllä itse hostatuissa open source vaihtoehtoisissa
- + Palveluntarjoajan tuki tuotteelle

Huonoa

- Kaikkia palvelun osia eikä tietoturvaa voida täysin itse hallita
- Pilvessä hostatut palvelut ovat usein maksullisia ja hinnoiteltu käyttäjäperusteisesti

XMPP-standardin mukainen itsehostattu palvelin

Kyseiseen standardiin perustuvien palvelimien edut ovat monessa kohdin samankaltaiset kuin muissa open source -chat-palvelimissa. XMPP-standardiin perustuvat projektit eivät kuitenkaan ole ehkä yhtä optimoituja kehittäjärajapintojen osalta ja myöskään monet palveluista eivät ole yhtä aktiivisen kehittäjäyhteisön tukemia (kaupallisen toimijan puuttuessa taustalta).

- Openfire, Ejabberd, Metronome, MongooselM, Tigase

Hyvää

- + Enimmäkseen samat kuin ei XMPP-standardia noudattavissa palveluissa

Huonoa

- Ei-kaupallisten projektien kehityksen jatkuvuus on usein arvaamattomampaa
- Ei välttämättä löydy yhden luokun ratkaisua jos standardissa määritellyt toiminnallisuudet eivät riitä

Chat-toiminnallisuuksien tekeminen itse alusta

Tässä vaihtoehdossa on toteutukseen vaadittavat resurssit luonnollisesti suurimmat. Itse toteutettavaan chat-palveluihin löytyy monia kevyitä arkkitehtuuriratkaisuja, mutta luotetun ja tietosuojaltaan hyvän palvelun toteuttaminen vaatisi moninkertaisen määrän resursseja tämän hankkeen tunnettujen tarpeiden osalta.

Hyvää

- + juuri halutut toiminnallisuudet eikä mitään ylimääräistä

Huonoa

- viestitoiminnallisuus, kanavat, tietoturva joudutaan kaikki tekemään alusta, joten kehitysresursseja menee perusasioihin huomattava määrä

Client-puolen toiminnallisuus

Käyttäjärajoituksissa chat-toiminnallisuuden ohjelmistokehitystä voidaan tehostaa käyttämällä räätälöityjen chat-sovellusten rakentamiseen erityisesti koottuja kirjastoja. Tässä erona palvelin-frameworkkeihin on ettei kypsiä open source -vaihtoehtoja oikeastaan löydy lainkaan. Kaupalliset versiot ovat todella monipuolisia ja tehokkaita kehittämisen kannalta, mutta niihin liittyy usein lisenssihinnoittelu, joka ei välttämättä toimi tämän hankkeen puitteissa. Asiakaspuolen chat-toiminnallisuudessa on myös näistä kirjastoista vähäisempi hyöty varsinkin jos käyttöliittymän visuaalisuutta halutaan hallita laajasti.

Hyvää

- + Nopea ottaa käyttöön
- + Jokseenkin räätälöitävissä

Huonoa

- Hinnoittelu
- Backend + Clientin yhteensopivuuteen täytyy kiinnittää huomiota

Oma räätälöity toteutus

Hyvää

- + Lean-toteutus vain tarvittavilla toiminnallisuuksilla
- + Helposti räätälöitävissä
- + Jatkokehitys helppoa
- + 100% varmuus toteutuksesta

Huonoa

- Enemmän työtä
- Hitaampi ottaa käyttöön

Bottitoiminnallisuudet

Tähän mennessä kaikki järjestelmään suunnitellut botin kanssa tehtävät vuorovaikutukset ovat teknisen toteutuksen kannalta normaalia staattista sisältöä, yksinkertaisia toimintoja, tiedon välitystä ja linkkejä. Tällöin botin rooli muodostuu käyttöliittymän selkeyttämisen ympärille hyvän suunnittelun kautta. Konseptissa botti toimii ikään kuin keskitettynä apurina, joka osaa vielä tarjota kunkin sivun mukaisia relevantteja toimintoja. Botin toiminnallisuuksien suunnittelussa on kuitenkin hyvä käyttää rakenteen selkeyden kannalta keskustelemaan vuorovaikutukseen perustuvia kaavioita, kuten *conversation mappeja*, *conversation diagrammeja* tai *decision treeitä*. Nämä saattavat osoittautua hyödyllisiksi varsinkin, jos chat-toiminnallisuuksiin lisätään myöhemmässä vaiheessa aitoa *conversational interface* toiminnallisuutta.

Tällä hetkellä aitoa älykästä vuorovaikutusta käyttäjän ja botin välille ei ole suunniteltu. Kuitenkin tunnettujen kehitystarpeiden kontekstissa on syytä pitää mielessä aidon chatbot-toiminnallisuuden mukanaan tuoman työmäärän mahdollinen lisäys suhteessa ominaisuuksien tarpeellisuuteen. Toisin sanoen, aidosti keskustelevan botin tarvetta ei alkuvaiheessa nähty, mutta tällainen tarve saattaa tulevaisuudessa syntyä. Mikäli tähän päädytään, olisi muut toiminnallisuudet ja niiden sijoittelu syytä miettiä uudelleen, sillä botin teknisestä toteutuksesta tulee erittäin monimutkainen, mikäli kaikki olemassaolevat toiminnot pyritään sisällyttämään sen toimintalogiikkaan.

Muita keskeisiä toiminnallisuuksia

Profiilin yhteyteen määritetyt kysymystoiminnallisuudet

Hankkeen keskeisimpiä onnistumista määrittäviä tekijöitä on toteutettavan palvelun kyky luoda aitoja kohtaamisia ihmisten välille. Monet tuntemistamme sosiaalisten medioiden palveluista keskittyvät ylläpitämään jo olemassa olevaa sosiaalista verkostoa. Tässä palvelussa halutaan toiminnallisuuksien puolesta löytää niitä tekijöitä, joilla suunnittelun ja teknisen toteutuksen puolesta voitaisiin edesauttaa kahdenvälisen yhteyksien ja keskustelun syntyä. Suunnittelutyön ja käyttäjähaastatteluiden perusteella profiilin yhteydessä toteutettava kysymystoiminnallisuus (päivän kysymykset) nähtiin tärkeänä edellä mainittua tavoitetta tukevana toiminnallisuutena.

Verso kaverilistassa

Siinä missä päivän kysymykset luo kohtaamisia, niin verso-toiminnallisuuden nähtiin pelillisenä elementtinä tukevan uusien kaveruuksien vuoropuhelun jatkumista aktiivisena tuttavuuden alkuvaiheilla. Verso on siis kahdenkeskisten vuoropuheluiden kasvattama visuaalinen elementti, joka voi hyvin kun keskustelu on aktiivista ja toistuvaa. Eli kun kaverit keskustelelee niin verso kasvaa ja kutsuu muistuttaen takaisin kahdenkeskisen vuoropuhelun äärelle.