**JOTU 2023, Määrittelydokumentaatio**

R140-JOTU22- maar1-v1

Softavirtuoosit Oy (Ryhmä 140)

Felix Karhusaari  
Opiskelijanumero: 150892825

Marika Bergman  
Opiskelijanumero: {opiskelijanro}

Tero Turja  
Opiskelijanumero: {opiskelijanro}

Jere Nieminen  
Opiskelijanumero: {opiskelijanro}

**Versiohistoria**:

* versio 0.1 | 09.03.2023 | pohjan luominen | Felix Karhusaari
* versio 0.2 | 09.03.2023 | 2.1, 2.2 | Tero Turja
* versio 0.3 | 09.03.2023 | 1.1 | Felix Karhusaari
* versio 0.4 | 09.03.2023 | 2.5 | Tero Turja
* versio 0.5 | 09.03.2023 | 1.4 | Felix Karhusaari

**Sisällysluettelo**:

* Luvut ja kohdat alla olevan mukaisesti numeroituna + sivunumerot. Muista että ”viimeinen tehtävä ennen palautusta” on generoida sisällysluettelo Wordin työkalulla (tai: ”Error… reference not found”).

# Johdanto

## Yleiskuvaus (tämän dokumentin tarkoitus ja sisältö)

Tämä dokumentti on tehty SähkönSäästö Appiksen (SÄSÄ) tarkempaa määrittelyä varten. Dokumentti on suunnattu niin tuotteen suunnittelutiimille asiakasyritykselle kuin sijoittajille. Määrittelydokumentti toimii referessinä projektin edetessä. Sen avulla voidaan varmistaa, että sovelluksen eri osa-alueet ja toiminnot huomioidaan ja totetutaan halutulla tavalla.

Määrittelyn tuote tulee {yritykselle}, joka on suomalainen tekniikan alan edelläkävijä. {Yritys} on suuren kasvupotentiaalin omaava startup-yritys, joka pysyttelee ajan hermoilla ja tarjoaa innovatiivisia ratkaisuja moderneihin ongelmiin.

Määrittelydokumentaatio kattaa sovelluksen käyttäjät ja käyttökontekstin, keskeiset toiminnallisuudet sekä tekniset vaatimukset, käyttöliittymän, kokonaisvaltaisen kuvauksen järjestelmän toiminnasta sekä jatkokehitysajatuksia. Lisäksi dokumentti havainnollistaa sovelluksen toimintaa ja käyttäjäkokemusta esimerkkien avulla. Dokumentin ulkopuolelle rajataan tuotteen markkinointiin ja brändäykseen liittyvät strategiat. Määrittely on luonteeltaan alustava - sen tavoitteena on tuotteen saattaminen onnistuneeseen MVP-vaiheeseen (minimum viable product). Näin ollen dokumentti ei myöskään kata pitkäjänteistä käyttäjälähtöistä testausta.

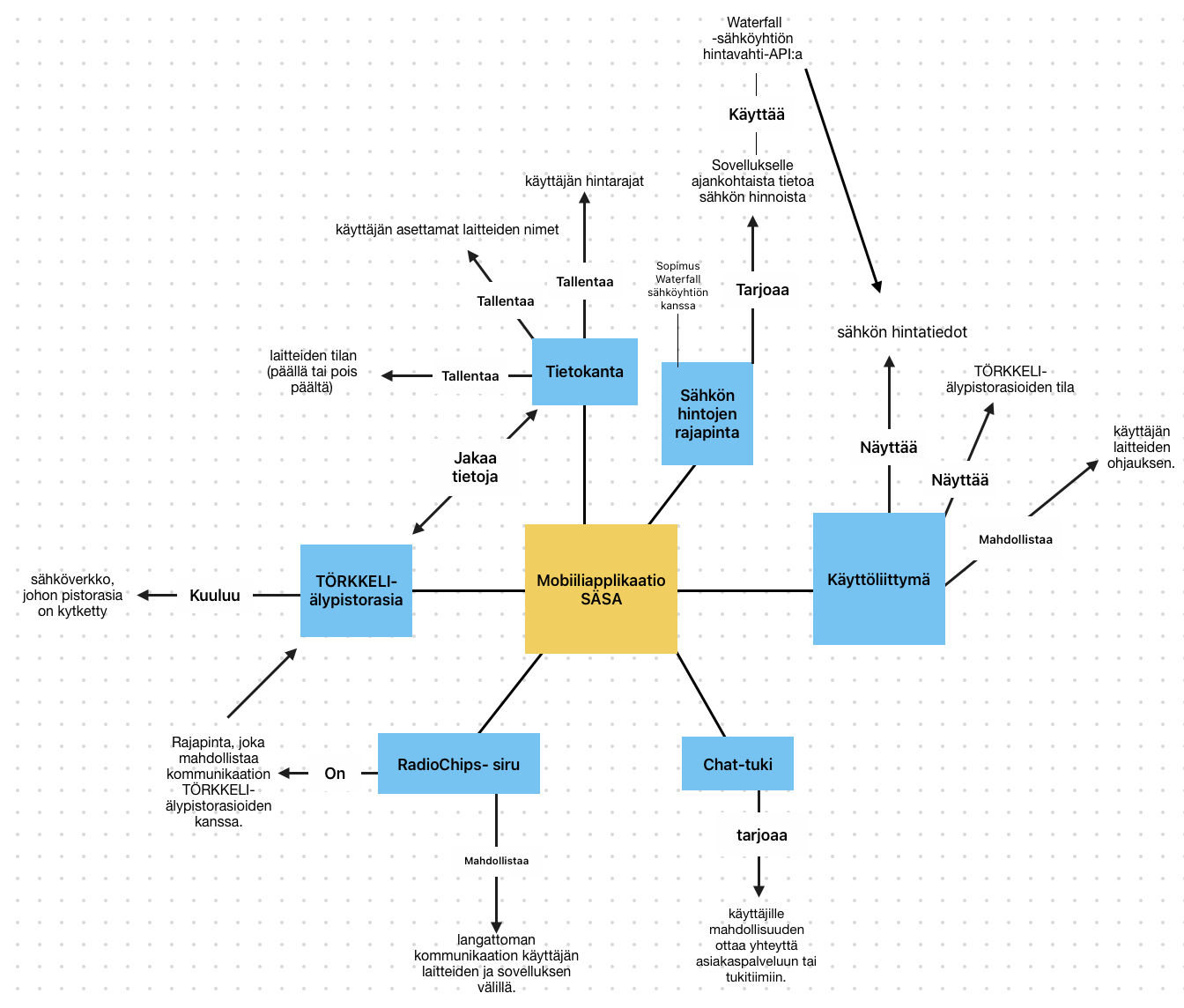
## Määriteltävä tuote, laajuus ja sen ympäristö

SähkönSäästö Appis on mobiilisovellus, joka on kehitetty TÖRKKELIN hallintaa varten. Sovelluksen avulla käyttäjä voi ohjata älypistorasioita ja seurata sähkön ajankohtaista markkinahintaa. Sovellus voi myös ilmoittaa käyttäjälle, jos sähkön hinta ylittää tai alittaa käyttäjän määrittämän rajan. RadioChips sirun avulla käyttäjä pystyy vaihtamaan laitteiden nimiä, mikä selkeyttää laitteiden ohjaamista, kun ei tarvitse muistaa monimutkaisia koodeja.

Sovellus auttaa käyttäjää hallitsemaan kodin sähkönkulutusta ja säästämään sähkölaskuissa. SÄSÄ:n avulla on helppoa ja nopea tarkistaa sähkön hinta. Tämä on kuluttajalle erityisen tärkeää tässä markkinatilanteessa, kun sähkön hinnat saavuttavat ennennäkemättömiä lukemia.

SÄSÄ:n laajuus on rajoitettu sähkön hinnan seurantaan sekä laitteiden nopeaan käynnistämiseen ja sammuttamiseen. SÄSÄ on tehty käyttäjiä varten, mikä näkyy esimerkiksi käyttäjäystävällisestä käyttöliittymästä. SÄSÄ:llä voi hallita mitä vain laitteita, kunhan TÖRKKELI on kytketty pistorasiaan. Sovelluksen tarjoamien graafien avulla käyttäjän on helppo saada kokonaiskuva sähkön hintojen vaihtelusta sekä omasta sähkönkulutustaan. Sovellus kattaa myös käyttäjän hallinnan, joka on toteutettu huomioiden yksityisyyteen ja tietoturvaan liittyvät hyvät käytännöt. Tämä pitää sisällään muun muassa mahdollisuuden kaksivaiheisen tunnistautumisen lisäämiseen.

SÄSÄ ei kuitenkaan automatisoi laitteiden toimintaa, vaan käyttäjän on itse ohjattava omia laitteitaan sovelluksella. On myös syytä huomioida, että Waterfallin API seuraa ainoastaan suomalaisia sähkön hintoja. SÄSÄn potentiaali käyttäjälle on suuri, mutta viime kädessä käyttäjä tekee sähkön kulutukseen liittyvät valinnat itse.



## Käyttäjät ja käyttötarkoitus

Kuvaillaan järjestelmän erilaiset käyttäjät (ketkä käyttävät järjestelmää) ja käyttötarkoitus yleisesti (toiminnot kuvataan tarkemmin 3. luvussa).

Käyttäjien (esim. loppukäyttäjä) ja heidän käyttöympäristöjen kuvaus. Käyttäjien mahdollinen ryhmittely.

Mahdolliset ylläpitäjät (pääkäyttäjä, ylikäyttäjä, admin)?

Käyttäjien asema organisaatiossa, koulutus, käyttö (päivittäin? puolivuosittain?)

Ketkä ovat tärkeimmät käyttäjät? Mitä käyttäjiltä oletetaan (osaaminen)? Tarvitsevatko tietyt käyttäjät koulutusta järjestelmän käyttöön?

Onko kyseessä yhden vai monen käyttäjän järjestelmä (systeemi)?

Käyttöliittymän kielivalinnat. Mahdollisesti käyttöliittymän poikkeukselliset piirteet.

## Määritelmät, termit ja lyhenteet

Tämän osion tarkoituksena on selventää mahdollisesti epäselviä käsitteitä ja lyhenteitä, jotka liittyvät SÄSÄ Appikseen ja sen kehittämiseen:

Tuote

* SÄSÄ / SÄSÄ Appis: SähkönSäästö Appis, eli tämän vaatimusmäärittelyn kohteena oleva sovellus
* TÖRKKELI: Älypistorasia, jonka ohjaamista varten SÄSÄ kehitetään
* Asiakas / Asiakasyritys: TÖRKKELIn kehityksen taustalla oleva yritys.
* {Yritys}: TÖRKKELIN kehityksen taustalla oleva yritys (ts. asiakasyritys)
* RadioChips / RadioChips™: Yritys, joka tarjoaa Radiochips sirun ja sen API:n törkkelin käyttöön.
* Waterfall: Yritys, joka tarjoaa sähön hintavahti-API:n sovelluksen käyttöön.
* Voima: Elektroniikkalaitteita myyvä kansainvälinen ketju, joka on luvannut ottaa tuotteen valikoimiinsa.
* Voimauttajat: Voiman myynti- ja myynnin tukihenkilöstö, jotka tarjoavat asennus- ja käyttötukea asiakkaille, jotka ovat ostaneet TÖRKKELIn heidän myymälästään.
* Spot-hinta: Sähkön markkinahinta tiettynä ajankohtana.

Tekninen sanasto

* Loppukäyttäjä / End user: Kuka tahansa, joka päätyy käyttämään lopullista SÄSÄ Appista.
* API: Teknologia, jonka avulla eri sovellukset tai tietojärjestelmät voivat kommunikoida keskenään ilman jatkuvaa manuaalista ohjausta. API:n avulla voidaan luoda yhteyksiä eri ohjelmistojen väille. Esimerkkinä Waterfall-yhtiön tarjoama hintavahti-API, josta SÄSÄ hakee käyttäjälle tietoa sähkön hinnoista.
* Käyttöliittymä: Vuorovaikutuspinta, joka toimii välikätenä teknisen sovelluksen ja käyttäjän välillä. Pitää sisällään esimerkiksi graafisen käyttöliittymän, joka kattaa mm. valikot, painikkeet, tekstikentät ja kuvakkeet.
* MVP: Minimum Viable Product, eli kehitettävän tuotteen versio, joka täyttää kaikki perustoiminnallisuudet.
* RadioChips siru: TÖRKKELIiin rakennettu siru ja sen hyödyntämä API, joka mahdollistaa TÖRKKELIn ja SÄSÄ Appiksen keskenäisen kommunikoinnin.
* Käytettävyys: Loppukäyttäjän sulavan ja toivotun käyttökokemuksen huomioiminen suunnittelussa
* Saavutettavuus: Loppukäyttäjän mahdollisten erityistarpeiden tai rajoitusten (esim. näköön liittyvät rajoitteet) huomioiminen suunnittelussa
* Käyttäjälähtöinen testaus: Sovelluksen testaaminen mahdollisilla loppukäyttäjillä. Auttaa käyttäjäperspektiivin ymmärtämisessä ja sovelluksen mahdollisten kipukohtien tunnistamisessa.
* Toiminnallinen vaatimus (T): Kuvaa sovelluksen toiminnallisuutta - esimerkiksi vaatimus siitä, että käyttäjä voi kirjautua sisään sovellukseen.
* Ei-toiminnallinen vaaatimus (E): Sovelluksen ominaisuudet, jotka eivät suoraan liity sen toiminnallisuuteen - esimerkiksi vaatimus sovelluksen helppokäyttöisyydestä.
* Reunaehto (R): Rajoitus tai ehto, joka voi vaikuttaa sovelluksen suunnitteluun tai toteutukseen - esimerkiksi tietoturvaan liittyvät vaatimukset.

# Vaatimusten keruusuunnitelma

Osana ensimmäistä vaihetta suunnitellaan, miten vaatimukset saadaan kerättyä tarpeellisilta sidosryhmiltä, ja tehdään sidosryhmäanalyysi.

## Taustatilanne

Mikä on lähtötilanne? Mitä uusi järjestelmä tuo uutena / korvaa / parantaa? Minkä ongelman tuote ratkaisee?

## Nykyisen dokumentaation ja vastaavien tuotteiden analyysi

Mitä dokumentaatiota projektista on tällä hetkellä saatavilla? Mitä sen perusteella tiedetään? Millaisia vastaavanlaisia (kilpailevia) tuotteita on olemassa? (Projekteissa ensimmäinen asia onkin tehdä verkkohaku, mitä samanlaisia tai vastaavia järjestelmiä on jo olemassa – julkisista lähteistä matkiminen on aina sallittua.)

## PESTE

PESTE-analyysi projektista (tunnista, miten projektin ympäristö voi vaikuttaa projektiin PESTE:n eri osa-alueilla, löydä sidosryhmät). Hyödynnä tämän analyysin tuloksia sidosryhmien tunnistamisessa.

## Sidosryhmäanalyysi

Sidosryhmäanalyysin tulokset, esitysmuotona taulukko. Sidosryhmät järkevästi luokiteltuna.

Sidosryhmien luokittelu, rooli, millä menetelmillä sidosryhmältä kerätään vaatimukset, sidosryhmän perustelu (miksi tarvitsemme tätä sidosryhmää?) ja tarvittava osallistuminen (milloin sidosryhmää tarvitaan?). Lisäksi luokittelu siitä, minkälaisia/mihin liittyviä vaatimuksia sidosryhmältä odotetaan saatavaksi.

Tästä pohja tiedostossa JOTU\_Sidosryhmä\_esimerkki.pdf (Huom, esimerkki on vain formaatista, sisältö viittaa täysin erilaiseen projektiin. Esimerkkitiedostojen sisällöt ovat myös keskenään eri projekteista. Omassa työssänne tulosten pitäisi olla yhteensopivia keskenään.)

## Alustavat vaatimukset ja niiden luokittelu

Annetun kehyskertomuksen pohjalta tunnistetut vaatimukset. Vaatimusten luokittelu, lähde ja tärkeys (priorisointi). Vaatimusten esitysmuotona taulukko on selkein. Tästä pohja tiedostossa JOTU\_Vaatimukset\_esimerkki.pdf (saa muokata vapaasti).

## Vaatimusten keruuprojektin suunnitelma

Vaatimusten keruuprojektin suunnitelma. Kuka tekee, mitä tekee ja koska. Esimerkiksi: Sidosryhmältä A vaatimusten kerääminen aivoriihessä, ajankohtana B, kesto C ja projektiryhmän D ja E osallistuvat. Kyseessä ei siis ole suunnitelma siitä, miten opiskelijoina toteutatte harjoitustyön.

Minkälaisia keruumenetelmiä käytetään vaatimusten keräämiseen? Sisällytä myös lyhyt kuvaus siitä, mitä käyttämänne keruumenetelmät tarkoittavat.

Aikatauluttakaa miten vaatimuksia kerättäisiin sidosryhmiltä. Älkää unohtako varata analysointiin aikaa. Kuka tekee, mitä ja milloin. Tavoitteena muodostaa selkeä kuva aikataulusta ja vastuista.

Esitysmuotona Gantt-kaavio (janakaavio). Gantt-kaavio, kuten kaikki kaaviot, voi hyötyä selittävästä tekstistä (esimerkiksi että tietynlaisilta sidosryhmiltä kerätään vaatimukset tietyllä tavalla).

# Liitteet (A, B, C, …)

Jos on tarpeen. Valmiissa määrittelyssä esim. asiakkaan käyttöliittymäohje tai rajapintakuvaus (open data API), tietokannan looginen kuvaus, tai jopa järjestelmän rakenteen suunniteltu arkkitehtuurikuvaus.