

# **Puutarha-aiheisen mobiilisovelluksen suunnittelu**



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinna, Tietojenkäsittelyn koulutusohjelma

Syksy, 2020

Hanna Söderberg

Tietojenkäsittelyn koulutusohjelma  
Hämeenlinna

---

<b>Tekijä</b>	Hanna Söderberg	<b>Vuosi</b> 2020
<b>Työn nimi</b>	Puutarha-aiheisen mobiilisovelluksen suunnittelu	
<b>Työn ohjaaja/t</b>	Erkki Laine	

---

## TIIVISTELMÄ

Opinnäytetyön tavoitteena oli luoda prototyyppi puutarha-aiheiselle mobiilisovellukselle. Työ toteutettiin projektina ilman toimeksiantoa omasta kiinnostuksesta käyttöliittymäsuunnittelua ja puutarhasovelluksia kohtaan.

Opinnäytetyön alussa on teoreettinen osuus, jossa käydään läpi ketterän kehityksen mukainen suunnittelumenetelmä, *design sprint*, sekä sovellussuunnittelun määrittelyvaihe keskittyen toiminnallisiin vaatimuksiin. Teoriaosuuden lopussa esitellään käyttöliittymäsuunnittelun periaatteita.

Käytännön osuudessa tutkitaan tekoälyä kasvintunnistussovellusten takana, vertaillaan eri kasvintunnistussovelluksia sekä esitellään käyttöliittymäprototyypin suunnitteluprosessi. Työn lopussa esitellään valmis prototyyppi ja vaatimusmäärittelyt.

**Avainsanat** Sovellussuunnittelu, mobiilisovellus, käyttöliittymä, prototyyppi, kasvintunnistus

**Sivut** 32 sivua, joista liitteitä 4 sivua

Degree Programme in Business Information Technology  
Hämeenlinna

---

<b>Author</b>	Hanna Söderberg	<b>Year</b> 2020
<b>Subject</b>	Designing a garden-themed mobile application	
<b>Supervisors</b>	Erkki Laine	

---

ABSTRACT

The aim of the thesis was to create a prototype for a garden-themed mobile application. The work was carried out as a project without a mandate, out of own interest to user interface design and gardening applications.

At the beginning of the thesis there is a theoretical part, in which the agile Design sprint method is reviewed. The theoretical part also includes a chapter about requirements specification. At the end of the theoretical part, the principles of user interface design are introduced.

The practical part starts with examining the artificial intelligence behind plant identification applications. After that the work compares different plant identification applications and presents the process of designing and creating a user interface prototype for the mobile application. At the end of the work, the finished prototype and functional requirements specifications are presented.

**Keywords** Software design, mobile app, UI, prototype, plant identification

**Pages** 32 pages including appendices 4 pages

# SISÄLLYS

1	JOHDANTO.....	1
2	SUUNNITTELUMENETELMÄN KUVAUS.....	2
3	SOVELLUSMÄÄRITYKSET.....	3
3.1	Kohderyhmä ja käyttäjäpersoona .....	3
3.2	Käyttötapaus .....	4
3.3	Toiminnalliset ja ei-toiminnalliset vaatimukset .....	4
4	MOBIILISOVELLUKSEN KÄYTTÖLIITTYMÄSUUNNITTELU.....	6
4.1	Käytettävyys ja heuristiikka.....	7
4.2	Visuaalisuus .....	8
5	KASVIEN TUNNISTAMINEN TEKOÄLYN AVULLA .....	10
6	KASVITIEKANNAT .....	12
7	SOVELLUSTEN VERTAILU .....	14
8	DESIGN SPRINT .....	16
9	SOVELLUKSEN MÄÄRITYSDOKUMENTAATIO .....	18
10	PROTOTYYPIN ESITTELY .....	20
11	YHTEENVETO .....	25
	LÄHTEET .....	26

## Liitteet

Liite 1	KÄYTTÖTAPAUKSET
Liite 2	EI-TOIMINNALLISET VAATIMUKSET

## 1 JOHDANTO

Kasveista ja luonnosta innostuneille ihmisille on tarjolla useita mobiilisovelluksia, joiden ideana on valokuvan perusteella tunnistaa, mikä kasvi tai eläin kuvassa on. Osa sovelluksista toimii käyttäjien lisäämien havaintojen ja tunnistusten perusteella. Yhteisön apu auttaa sovelluksen kykyä tunnistaa lajeja kehittymään. Kasvintunnistussovellus PlantSnapin taustalla toimiva tekoälyalgoritmi on oppinut tunnistamaan satoja tuhansia lajeja.

Tunnetuimpia kasvintunnistussovelluksia ovat PlantNet, PlantSnap, PictureThis ja iNaturalist. Lisäksi Google Lens toimii hyvin myös kasvien tunnistamiseen. Kappaleessa 7 vertaillaan maksuttomia puutarha- ja kasvintunnistussovelluksia ja arvioidaan, mikä sovelluksissa on aloittelijan näkökulmasta hyvää ja mitä voisi edelleen parantaa.

Tunnistusominaisuuksien lisäksi aloitteleva puutarhuri voi kaivata myös vähemmän tarjolla olevia sovelluksia, jotka ohjeistaisivat pihan hoidossa vuodenkierron eri aikoina. Osa sovelluksista, kuten Blossom, pääsee melko lähelle tätä tarvetta, vaikkei sekään täysin täytä odotuksia. iPhoneille ei ainakaan vielä ole löytynyt ohjeita antavaa sovellusta, joka toimisi hyvin myös suomen kielellä.

Opinnäytetyöprojektissa suunnitellaan yksinkertainen puutarha-aiheinen käyttöliittymä mobiililaitteilla käytettäväksi sekä määritellään toiminnalliset vaatimukset. Käyttöliittymässä on mukana kuvasta tunnistaminen sekä ohjeistava kalenteritoiminnallisuus. Työ on tehty ilman toimeksiantoa omien kiinnostuksen kohteiden pohjalta.

Opinnäytetyön teoreettisessa viitekehyksessä käsitellään ketterän kehityksen mukaista Design Sprint -menetelmää, mobiilisovelluksen määrittämiä ja käyttöliittymäsuunnittelua sekä kuvatunnistuksen teknologiaa. Käytännön osuudessa on sovellusvertailun lisäksi esitelty prototyypin suunnitteluprosessi ja vaatimusmäärittäykset sekä itse prototyyppi.

Opinnäytetyössä etsitään vastauksia seuraaviin tutkimuskysymyksiin:

- Kuinka varmistaa mobiilisovelluksen helppokäyttöisyys?
- Millaisia kasvitietokantoja on saatavilla?
- Kuinka kuvasta tunnistaminen toimii?

## 2 SUUNNITTELUMENETELMÄN KUVAUS

Ketterä kehitys on yleistermi erilaisille joustaville metodologioille ja työskentelytavoille. Ketterässä kehityksessä monialaiset tiimit toimivat itsenäisesti ja tekemisessä korostuvat yhteistyö ja itsensä johtaminen. Tiimit voivat itsenäisesti päättää, kuinka työt hoidetaan monialaista osaamista hyödyntäen. Metodologia on kokoelma tapoja, joihin tiimi on yhdessä sitoutunut. Esimerkiksi Scrum on eräänlainen metodologia, joka on yleisesti käytössä mutta jokainen tiimi soveltaa omalla tavallaan sen viitekehystä. (Agile Alliance, 2020)

Sprint on ketterän kehityksen tapahtuma, joka on keskeinen nimenomaan Scrumissa. Scrumissa sprintit kestävät kolmesta neljään viikkoa. Sprintin aikana kehitetään tavoitteen mukainen, ainakin lähes valmis ratkaisu. Sprintiin sisältyy suunnittelu, tavoite (sprint goal), kehitystyö, päivittäiset palaverit (daily scrums) sekä sprintin arviointi. (Schwaber, K. & Sutherland, J., 2017)

Samanlaisia tavoitteita ja prosesseja käytetään myös viisipäiväisessä Design Sprintissä, joka valittiin myös tämän opinnäytetyön työskentelymenetelmäksi. Design sprintillä on mahdollista saada nopeasti aikaan toimiva prototyyppi, sillä koko prosessi on läpikäyty alle viikossa. Design sprint on Google Venturesin luoma menetelmä, jolla ratkotaan liiketoiminnan ongelmia sekä luodaan uusia tuotteita ja parannellaan nykyisiä. Menetelmä on yleisesti käytössä tuotekehityksessä ja palvelumuotoilussa. (Meom, 2020)

Design sprintin ensimmäisenä päivänä eli maanantaina kartoitetaan ongelmat ja valitaan keskittymisalueet. Ensimmäisen päivän aikana sovitaan myös pitkän aikavälin tavoite. Näiden asioiden ratkaisemiseksi piirretään kartta käyttäjän polusta, jota täydennetään viikon aikana. Aluksi kartalle merkitään käyttäjä(t) ja muut tärkeät avaintahot. (GV, 2019)

Toisena päivänä piirretään paperille erilaisia rautalankaversioita suunnitellusta prototyypistä. Tiistaina tehdään päätöksiä siitä, mihin suuntaan prototyyppiä tulisi viedä. Tässä vaiheessa on tärkeää huomioida, kuinka vastaavanlaiset sovellukset on aiemmin toteutettu. (GV Library, 2016)

Kolmantena päivänä keskitytään päätöksentekoon ja muokataan ideat testattaviksi hypoteeseiksi. Neljäntenä päivänä toteutetaan korkealuokkainen prototyyppi ja viidentenä päivänä, joka yleisesti on perjantai, valitut käyttäjät testaavat prototyyppiä. (GV, 2019)

### 3 SOVELLUSMÄÄRITYKSET

Käyttöliittymälähtöinen määrittely voi olla järkevä vaihtoehto sille, että yritetään heti projektin alussa määritellä sovelluksen toiminnallisuudet mahdollisimman valmiiksi. Kehitystyö voi lähteä helposti virheelliseen suuntaan ja pahimmassa tapauksessa tehdään jotain, mitä kukaan ei tahdo käyttää. Käyttöliittymälähtöisessä määrittelyssä määritellään ensin, miltä sovellus näyttää ja lopputulos päästään esittelemään asiakkaalle heti projektin alkuvaiheessa. Tällä tavoin voidaan jo heti alussa löytää puutteita ja ongelmakohtia suunnitelmasta. (Laakso, 2006.)

Opinnäytetyössä luotavan prototyypin kannalta oleellisimpia vaiheita ovat käyttöä kuvailevat määrittelyt eli käyttötapaukset ja toiminnalliset vaatimukset. Toiminnalliset vaatimukset ovat määrittelyjä, jotka loppukäyttäjä ehdottomasti haluaa sovellukseen mukaan. Jotta toiminnalliset vaatimukset voidaan tarkasti kuvata, tulee kaikki mahdolliset skenaariot luetella. Toiminnalliset vaatimukset voidaan kuvata joko luonnollisella kielellä tai muodollisemmilla tavoilla, kuten UML-kaavioilla. (Geeksfor-Geeks, 2018.)

#### 3.1 Kohderyhmä ja käyttäjäpersoonat

Loppukäyttäjien tuominen lähemmäksi kehitystiimiä on ensisijaisen tärkeää lopputuloksen onnistumisen kannalta. Etenkin ohjelmistotiimit, jotka tuottavat sovelluksia kuluttajille, kohtaavat usein ongelmia suurten käyttäjäryhmien kanssa kommunikoidessa. Kehityssyklit ovat nopeita ja tarkoituksena on jatkuvasti parantaa tuotteita ja tyydyttää mahdollisimman monen asiakkaan tarpeet. (Alvertis ym., 2016, s. 73-83.)

Sovelluskehittäjien ja asiakkaiden välistä yhteistyötä parantamaan on kehitteillä alustoja, joiden avulla asiakaskunta on helposti integroitavissa kehitysprosessiin. Sosiaalisen median ja erilaisten crowdsourcing-alustojen käyttö auttaa pienempiäkin tiimejä integroimaan asiakasrajapinnan kehitystyöhön. Erilaisissa joukkoistamis- ja joukkorahoituskampanjoissa voidaan kerätä palautetta ja rahoitusta potentiaalisilta käyttäjiltä. (Alvertis ym., 2016, s. 73-83.)

Asiakkaiden osallistuminen sovelluskehitykseen on yleensä maltillista ja on hankalaa tietää, toimiiko kehitettävä sovellus suurelle yleisölle. Kehitteillä onkin jo työkaluja ja alustoja, joilla loppukäyttäjien ja kehittäjien yhteistyötä pyritään parantamaan. Käyttäjät eivät kuitenkaan aina itse ymmärrä omia tapojaan tai tarpeitaan. Siksi heidän seuraamisensa voi hyödyttää kehitystiimiä suuresti. Jos käyttäjistä on saatavilla dataa, sitä kannattaa käyttää user personan eli käyttäjäpersoonan luomiseksi. (Alvertis ym., 2016., s. 73-83)

Palvelumuotoilu on käytäntö, jolla suunnitellaan palveluja kokonaisvaltaisesti ja yhteistyöhön perustuen. Palvelumuotoilun tarkoituksena on tuottaa arvoa sekä loppukäyttäjälle että palveluntarjoajalle koko palvelun olemassaolon ajan. (Service Design Network, 2020)

Palvelumuotoiluprosessiin kuuluva käyttäjäpersoonana on keksitty henkilö, jolle sovellusta lähdetään tekemään. Persoonat ovat oikeaan dataan perustuvia hahmoja, joiden käyttäminen auttaa ymmärtämään sovelluksen erilaisia käyttötapoja ja kipupisteitä. (Friis Dam & Yu Siang, 2020.)

### 3.2 Käyttötapaus

Käyttötapaus on kuvaus tapahtumasta, jonka käyttäjä suorittaa ohjelmistolla. Käyttötapaukset kuvaavat sitä, miten ohjelmiston tulisi toimia käyttäjän näkökulmasta, ja auttavat hahmottamaan mahdollisia virhetilanteita. Käyttötapauksen perusteella voidaan arvioida sovelluksen monimutkaisuutta ja hintaa sekä muodostaa vaatimusmääritykset. (Usability.gov, 2020)

Käyttötapauksen kuvaamiseen käytetään UML-kaaviota, jota voidaan avata sanallisella taulukkomuotoisella käyttötapauskuvauksella. UML eli Unified Modeling Language on standardoitu esitystapa, jonka tarkoitus on esittää asia niin, että kaikki sitä lukevat ymmärtävät asian samalla tavalla. UML-kaavion käyttäminen voi selkeyttää määrityksien esittämistä mutta kaaviot tulisi muistaa aina päivittää, kun ohjelmistoa päivitetään. Alkumääritysten jälkeen tällainen ei välttämättä ole kovin mielekästä, joten liian tarkalle tasolle käyttötapauksissa ei kannata mennä. (Leppäniemi, 2012)

Koska UML on standardoitu esitystapa, käyttötapauskaavio tulee piirtää tietyllä tavalla. Kaaviossa suunniteltava systeemi piirretään laatikoksi, jonka sisällä on soikioiden sisään kirjattu sovelluksessa tapahtuvat eri toiminnot eli käyttötapaukset (use case). Kaaviossa on oltava toimija (actor), josta lähtee viiva käyttötapaukseen. Käyttötapaukset voivat olla suhteessa toisiinsa, mikä merkitään katkoviivanuolella. Kun käyttötapaukseen voi kuulua lisäarvoa tuova toinen käyttötapaus, merkitään se sanalla <<extend>>. Käyttötapauksen sisällä tapahtuva erillinen käyttötapaus merkitään sanalla <<include>>.

### 3.3 Toiminnalliset ja ei-toiminnalliset vaatimukset

Kun käyttötapaukset on määritelty, voidaan määritellä toiminnallisia vaatimuksia. Toiminnallinen vaatimus kertoo, miten sovelluksen tulisi toimia. Toiminnalliseen vaatimukseen liitetään nimi ja id, lyhyt kuvaus ja perustelu, sovellukselta vaadittavat toiminnot sekä referenssinä olevat käyttötapaukset tai muut vaatimukset. (Stellman, A. & Greene, J., 2005, s. 111-112)



Toiminnallisen vaatimuksen tärkein osuus on halutun käyttäytymisen/ta-  
pahtuman kuvaaminen. Kuvaus voi sisältää listoja tai kuvia, kunhan lukija  
ymmärtää, mitä pitää tapahtua. Epäselvää kieltä ja monitulkintaisia sana-  
muotoja tulee välttää. Myös design-elementteihin viittaavia sanoja, kuten  
nappi tai valintaruutu, tulee välttää toiminnallista vaatimusta kirjoitta-  
essa. Suunnittelu tehdään vaatimusmääritysten jälkeen ja olisi suunnitte-  
lijän innovointikyvyn ja ammattitaidon aliarvioimista vaatia tietynlaista  
designia jo tässä vaiheessa. (Stellman, A. & Greene, J., 2005, s. 112-113)

Ei-toiminnalliset vaatimukset määrittelevät, miten helppoa tai nopeaa so-  
vellusta on käyttää, kuinka luotettava sovellus on ja kuinka hyvin se käyt-  
täytyy, kun odottamattomia asioita tapahtuu. Ei-toiminnallisia vaatimuk-  
sia kutsutaan joskus myös sovelluksen laatutekijöiksi. Ei-toiminnalliset  
vaatimukset pitäisi kuvailla niin tarkkaan kuin mahdollista, lukumääriä ja  
mittayksiköitä apuna käyttäen. Ei-toiminnalliset vaatimukset jaetaan eri  
kategorioihin, jotka on selitetty taulukossa 1. (Stellman, A. & Greene, J.,  
2005, s. 113-114)

Taulukko 1. Ei-toiminnallisten vaatimusten luokittelu.

Luokka	Kuvaus
Saatavuus (Availability)	Aika, jonka sovellus on käytettävissä.
Tehokkuus (Efficiency)	Kuinka hyvin sovellus käyttää resurssejaan, ku- ten CPU:ta ja levytilaa.
Joustavuus (Flexibility)	Kuinka sovellusta voidaan myöhemmin laajen- taa.
Siirreltävyys (Portability)	Vaatimukset sille, millä eri alustoilla sovelluksen tulee toimia.
Eheys (Integrity)	Tietoturvaan, tietosuojaan ja käyttöoikeuksiin liittyvät määritykset.
Suorituskyky (Performance)	Toiminnot ja ominaisuudet, joihin liittyy aikape- rusteisia rajoituksia.
Luotettavuus (Reliability)	Kuinka hyvin sovellus säilyttää suorituskykynsä.
Uudelleenkäytettävyys (Re- usability)	Sovelluksen kyky käyttää samaa ominaisuutta monessa paikassa.
Kestävyys (Robustness)	Sovelluksen kyky hallita virhetilanteita.
Skaalautuvuus (Scalability)	Millä tavoin sovellus on skaalautuva.
Käytettävyys (Usability)	Helppokäyttöisyyteen liittyvät vaatimukset.

## 4 MOBIILISOVELLUKSEN KÄYTTÖLIITTYMÄSUUNNITTELU

Käyttöliittymä (UI, user interface) on laitteen ja ihmisten välisen vuorovaikutuksen väline, jonka avulla ihminen ohjaa laitteen tai ohjelmiston tapahtumia. Tässä opinnäytetyössä tarkastellaan nimenomaan mobiililaitteiden graafisen käyttöliittymän suunnittelua.

Mobiililaitteet ovat kooltaan pieniä, minkä vuoksi niitä on myös kognitiivisesti rasittavampaa lukea kuin desktop-laitteita. Tämä tulee huomioida sovelluksen suunnittelussa siten, että käyttöliittymä on mahdollisimman selkeä ja tekstiä on vain vähän kerrallaan luettavana. Mobiilikäyttäjät ovat myös useammin kiireessä ja heidän muistinsa joutuu koetukselle heti, kun tieto rullautuu pois näkyvistä. Sovelluksen suunnittelussa tulee ottaa nämä asiat huomioon ja pitää sisältö erittäin yksinkertaisena. (Nielsen, 2011.)

Näytön pienuus tulee huomioida monin tavoin, esimerkiksi valikkojen ja muiden kiinteiden elementtien asettelulla. Valikot eivät saisi viedä liikaa tilaa näytöltä, jotta sisältö mahtuu paremmin näkyville. Näytön pienuus vaikeuttaa myös kirjoittamista ja näkemistä, joten tekstikenttiä kannattaa välttää tai ainakin pyrkiä auttamaan käyttäjää niiden täyttämässä. Kehotepainikkeet kannattaa tehdä suuremmiksi, mitä ne olisivat vastaavalla internetsivuilla, jotta niihin on helpompi osua sormella. (Budiun, 2019.)

Mobiililaitteilla on muitakin erityispiirteitä, jotka erottavat mobiilisovellukset esimerkiksi internetsivuista. Budiun (2019) mukaan mobiililaitteiden käyttö on nopeaa ja väliaikaista, selailukerta kestää keskimäärin vain 72 sekuntia ja käyttäjän huomio voi samaan aikaan olla myös toisaalla ympäristössä. Tämän vuoksi sovelluksen tulisi tallentaa aina käyttäjän tila ja viimeisimmät muutokset automaattisesti. Jos näyttö sammuu tai käyttäjä poistuu toiseen sovellukseen, pitäisi hänellä olla mahdollisuus jatkaa siitä, mihin jäi, kun hän taas palaa sovelluksen pariin. Sovellusta suunniteltaessa tulee huomioida polku, mitä pitkin käyttäjä pääsee tavoitteesensa mahdollisimman vähällä vaivalla. Tehtävien tulee olla yksinkertaisia valintoja mieluummin kuin monimutkaisia tekstinsyöttöjä ja tarkistuksia.

Kosketusnäyttö ja mobiililaitteiden muut ominaisuudet voivat tehdä sovelluksesta helpommin tai hauskemman käytettävän. Kosketusnäyttö luo kuitenkin myös virhenäppäilyjen mahdollisuuden, joten peruuttaminen ja virhelyöntien korjaaminen pitäisi aina olla mahdollista. Sovelluksen käyttöä voi helpottaa integroimalla siihen muita sovelluksia, kuten GPS-paikannusta ja kameraa, jota myös opinnäytetyön aiheena oleva prototyyppi käyttää. Esimerkiksi yhteystietojen kohdalla voi olla soitonappi, josta puhelu aktivoituu suoraan ilman että käyttäjän tulee erikseen valita puhelinsovellus ja näppäillä numero sinne. Push-ilmoituksilla käyttäjälle

on helppo antaa informaatiota ja kehoitteita palata sovelluksen pariin. (Budi, 2019.)

Käyttöliittymäsuunnittelua helpottaa Ben Shneidermanin kahdeksan kultaista sääntöä. Vaikka säännöt ovat 1980-luvulta, ne luovat edelleen hyvän pohjan web- ja mobiilisovellusten suunnittelijoille. Sääntöjä myös päivitetään nykypäivän tarpeisiin. (Shneiderman, n.d.)

Kahdeksan kultaista sääntöä ovat seuraavanlaiset:

1. **Jatkuvuuteen pyrkiminen:** Komennot ja kehoitteet ovat joka paikassa samanlaiset. Kieli ja terminologia on yhteneväistä.
2. **Universaaliin käytettävyyteen pyrkiminen:** Käyttöliittymän tulee olla yhtä lailla käytettävissä, vaikka käyttäjä olisi aloittelija tai kokenempi. Myös erilaiset käyttötavat ja käytön esteet tulisi huomioida, eli sovelluksesta pitäisi tehdä mahdollisimman hyvin saavutettava.
3. **Anna käyttäjälle informatiivista palautetta:** Jokaisesta käyttäjän toiminnosta tulee antaa palaute, esimerkiksi siirtää käyttäjä seuraavaan näkymään klikkauksen jälkeen.
4. **Kerro lopputulos käyttäjälle:** Käyttäjälle on hyvä kertoa, milloin hän on saanut tehtävän päätökseen. Tässä auttavat perinteiset Kiitos-sivut ja vastaavat kuittaukset.
5. **Estä virhetilanteet:** Tee käyttöliittymä sellaiseksi, etteivät käyttäjät pysty rikkomaan sitä. Mikäli virhe meinaa tapahtua, käyttöliittymän tulisi antaa rakentava palaute ja ehdotus toisenlaisesta menettelystä.
6. **Tee peruuttaminen helpoksi:** Käyttäjän toimintojen pitäisi aina olla peruttavissa. Käyttäjän tuntee helpotusta, kun voi perua virheellisen painalluksen.
7. **Anna käyttäjälle hallinnan tunne:** Käyttöliittymän ei pitäisi toteuttaa toimintoja, joita käyttäjä ei ole pyytänyt.
8. **Vähennä muistinvarassa toimimista:** Mitä vähemmän käyttäjän tarvitsee muistaa edellisistä vaiheista, sitä parempi.

#### 4.1 Käytettävyys ja heuristiikka

Käytettävyys on laadullinen mittari sille, miten helppoa käyttöliittymää on käyttää. Käytettävyydellä myös viitataan menetelmään, joka pyrkii parantamaan käytön helppoutta suunnitteluprosessin aikana. Käytettävyys määritellään viidellä laatutekijällä, joita ovat opittavuus, tehokkuus, muistettavuus, virheiden vähyys ja tyytyväisyys. Muitakin tekijöitä on, kuten hyödyllisyys eli se, tekeekö sovellus sitä mitä tarvitaan. Käytettävyys ja hyödyllisyys ovat molemmat tärkeitä, sillä niistä yhdessä muodostuu sovelluksen käyttökelpoisuus. (Nielsen, 2012)

Opittavuus mittaa, miten helposti käyttäjä pääsee lopputulokseen, kun hän käyttää sovellusta ensimmäistä kertaa. Tehokkuus mittaa tehtäviin kulutettua aikaa ja muistettavuus mittaa sitä, miten helposti käyttäjä muistaa sovelluksen, kun on ollut jonkin aikaa käyttämättä sitä. Virheet

mittaavat käyttäjän tekemiä virheitä, miten vakavia virheet ovat ja voiko virheen jälkeen palata takaisin. Tyytyväisyys mittaa sitä, miten miellyttävä sovellus on käyttää. (Nielsen, 2012)

Käytettävyys on erittäin tärkeää, sillä jos sovellus on hankala käyttää tai se ei anna sitä tietoa, jota käyttäjä etsii, käyttö lopetetaan (Nielsen, 2012). Tämä korostuu hektisessä mobiilisovellusmaailmassa, jossa tehtävät halutaan saattaa loppuun nopeasti ja vähällä vaivalla.

Heuristiikka on ongelmanratkaisussa käytettävä ajattelumenetelmä. 1990-luvulla kehitettiin käytettävyyden arviointiin heuristiset nyrkkisäännöt. Interaction Design Foundation (n.d.) kuvaa nämä Nielsenin ja Mollichin heuristisen arviointimenetelmän periaatteet seuraavasti:

1. Käyttäjälle näytetään sovelluksen tila oikea-aikaisesti ja tilanteeseen sopivalla tavalla.
2. Käyttäjälle näytetään tietoa vastaavasti kuin oikeassa maailmassa tapahtuisi.
3. Käyttäjälle annetaan hallinta ja helppo tapa perua toimintoja.
4. Sovelluksessa käytetään johdonmukaisesti ikoneja ja termejä.
5. Sovellus ohjeistaa välttämään virhetilanteet varmistamalla toiminto ennen toteutusta.
6. Käyttäjä saa tarvittavat ohjeet ja informaation käytön yhteydessä ilman muistinvaraisuutta.
7. Sovellus joustaa erilaisten käyttötapojen mukaan.
8. Sovelluksen designin tulisi olla vähäeleistä ja sisältää vain tarvittavat elementit.
9. Sovellus tarjoaa selkeäsanaisesti apua virhetilanteissa.
10. Sovellus tarjoaa selkeää apua ja dokumentaatiota.

Näitä periaatteita voidaan edelleen soveltaa myös mobiilikäyttöliittymien käytettävyyden arviointiin.

## 4.2 Visuaalisuus

Visuaalista estetiikkaa voidaan tarkastella monelta kantilta ihmisen ja tietokoneen välisessä vuorovaikutuksessa (HCI, human-computer interaction). Tractinskyn (2020) mukaan visuaalisuus ja muut suunnitteluperiaatteet kohtaavat toisensa hyvin. Estetiikka ei suinkaan huononna sovelluksen käytettävyyttä vaan päinvastoin tukee sitä. Monet käytettävyyttä parantavat periaatteet kuten standardinomaiset layoutit, elementtien asettaminen järjestykseen ja niiden toisistaan erottaminen on myös visuaalisesti esteettisempää. Esteettinen käyttöliittymä parantaa myös käyttäjäkokemusta, sillä estetiikka on ihmiselle miellyttävää ja parantaa hyvinvointia. "Kaunis" käyttöliittymä luo mielikuvan paremmasta käyttöliittymästä.

Viime vuosikymmenien menestyneimmät teknologiayritykset ovat siirtyneet tuottamaan kuluttaja- ja viihdetuotteita. Kuluttajakeskeisyyden ja interaktiivisten tekniikoiden hyödyntäminen lisää estetiikan merkitystä kilpailevista tuotteista erottavana tekijänä. Älypuhelimet ja tablettitietokoneet on suunniteltu parantamaan käyttäjäkokemusta, ja kuluttajan huomioon ja kiintymykseen pyritään ulkonäöllä ja muotoilupohjaisella symbolisella arvolla. Näin ollen myös esteettinen suunnittelu on alkanut saada hyväksyntää erottautumisstrategiana. (Tractinsky, 2020)

HCI-tutkimuksessa tarkastellaan ensisijaisesti visuaalisen estetiikan arvoa. Jo varhaisissa tutkimuksissa on havaittu estetiikan ja käytettävyyden välinen assosiaatio. Ihmisen käsitys järjestelmän kauneudesta voi vaikuttaa hänen käsitykseensä myös muista järjestelmän ominaisuuksista, kuten helppokäyttöisyydestä ja jopa suorituskyvystä.

## 5 KASVIEN TUNNISTAMINEN TEKOÄLYN AVULLA

Kasvintunnistussovelluksissa on suunnilleen sama käyttäjälle näkyvä prosessi, mutta taustalla toimiva tekoäly voi erota sovelluksien välillä. Tunnistusta varten käyttäjä ottaa mahdollisimman tarkan lähikuvan kasvista, minkä jälkeen sovellus analysoi kuvan konenäön (computer vision) ja koneoppimisen (machine learning) keinoin ja vertaa kuvaa tietokannassa oleviin. Tähän tapaan perustuu muun muassa ensimmäisiin julkaistuihin kasvintunnistussovelluksiin kuuluva LeafSnap.

Uusimmat sovellukset toimivat syväoppivien neuroverkkojen avulla, jolloin ne voivat tunnistaa satoja tuhansia kasveja. Syväoppiminen (deep learning) on koneoppimismenetelmä, jossa muodostetaan neuroverkon kerroksia siten, että tieto kulkee kerrosten läpi vuoron perään. Verkon kerroksellisuus mahdollistaa monimutkaisten rakenteiden oppimisen ilman valtavia datamääriä. (Reaktor, n.d.)

Brownleen (2019) mukaan konenäkö on tekoälyn ja koneoppimisen osa-alue, jolla tietokone pyrkii ymmärtämään ja analysoimaan kuvan sisältöä. Ensimmäisen kasvintunnistussovelluksen, LeafSnapin, toiminta perustuu hahmontunnistukseen, jossa on tavoitteena antaa lähimpänä todennäköisintä oleva vastaus. Sovellus antaa käyttäjälle lähimmät kolme vaihtoehtoa, joista käyttäjä itse valitsee mielestään oikean vastauksen.

Koska lehden muoto on sen pysyvin ominaisuus, tämä otettiin perustaksi LeafSnapin kehitykselle. Kehitystiimi aloitti määrittelemällä muotoja ja tekemällä vertailuja, minkä jälkeen he ohjelmoivat tietokoneen tekemään yksinkertaista eliminointia. Tiimi opetti koneen erottamaan lajeja toisistaan yksinkertaisilla erottimilla, jotta voidaan kertoa lehden muistuttavan enemmän jotain yhtä puulajia kuin muita. (Phys.org, 2011.)

Ensimmäisessä versiossa LeafSnap pystyi tunnistamaan puun kuvasta, jossa oli ainoastaan selkeästi kuvattu lehti vaaleaa taustaa vasten. Tunnistusprosessin ensimmäisessä vaiheessa kuva luokiteltiin lehdeksi tai joskin muuksi. Seuraavaksi kuva segmentoitiiin käyttäen tilastotieteellisiä keinoja erottamaan kohde taustasta. Segmentoinnin jälkeen arvioitiin lehden muoto laskemalla kuvan eri pikselivälien kaarevuuksia. Selvitettyjä ominaisuuksia (muoto ja väri) vertailtiin tietokannassa oleviin kuviin ja lähimmät vastaavuudet esiteltiin käyttäjälle. Tunnistusprosessi kesti alkuun muutaman sekunnin, jonka jälkeen käyttäjä voi itse valita oikean kasvin annetuista vaihtoehdoista. (Kumar, N. 2012)

Google Lens -sovellus puolestaan käyttää syväoppimiseen pohjautuvaa tekoälyä tunnistamaan kuvan kohteen ja hakemaan siitä tietoa. PlantSnap-sovellus käyttää neuroverkkoja tunnistamaan satoja tuhansia kasvilajeja kuvan perusteella. Syväoppiminen mahdollistaa sen, että enemmän

sovellusta käytetään, sitä paremmin se pystyy tunnistamaan kasveja. (MobileSyrup, 2018)

Mikäli kasvi tunnistetaan virheellisesti tai ei ollenkaan, PlantSnap-sovellusta voi opettaa parillakin tavalla. Kasville voi ehdottaa nimeä itse tai kuvan voi lähettää tunnistettavaksi toiselle käyttäjälle. Vastauksen saa minuuteissa. Jokaisella tunnistetulla kasvilla on oma sivu, jossa on perustietoja kasvista. (MobileSyrup, 2018)

PlantNet-sovelluksen kuvatunnistus käyttää konvoluutioneuroverkkoja (convolutional neural network, CNN), jota säännöllisesti opetetaan ohjautusti. PlantNetin tekoäly on opetettu alun perin ImageNetin datasetin perusteella ja myöhemmin sitä on jatko-opetettu PlantNetin omalla datalla. PlantNet tunnistaa yli 10000 lajia. (Affouard, A. ym., 2017)

## 6 KASVITIEKANNAT

Ulkomaisia kasvitietokantoja on paljon internetissä avoimesti saatavilla. Opinnäytetyön kohteena olevaan sovellukseen haluttiin kuitenkin etenkin kotimaisten kasvien tietämystä, joten tutkittavat tietokannat on keskitetty kotimaisiin, suomenkielisiin tietokantoihin.

Useimmat digitaalisesti saatavilla olevat kasvitietokannat ovat kasvioppaan tyyliin käytettäviä hakupalveluja. Yksi näistä on Luontoportti, joka on käytettävissä osoitteessa [www.luontoportti.fi](http://www.luontoportti.fi). Kasvien tunnistamista ohjaa web-käyttöliittymä, jossa tehdään valintoja liittyen kasvuolosuhteisiin ja kasvin ominaisuuksiin. Hakupalvelun käyttö vaatii hieman ymmärrystä kasveista ja kasvisanastosta. Englanniksi vastaavanlainen tietokanta löytyy muun muassa englantilaiselta The Royal Horticultural Societylta.

Helsingin yliopiston ja Luonnontieteellisen keskusmuseon Pinkka-oppimisympäristö on jaoteltu käyttäjäryhmien mukaan ”pinkoiksi”, jotka sisältävät kyseiseen ryhmään kuuluvia lajeja. Harrastelijaa voisi kiinnostaa esimerkiksi Pohjolan kasveja- tai Villivihannekset Suomen luonnossa -nimiset kokoelmat kasvilajeista, joita kasvaa Suomessa ja lähialueilla. Kasvin kuvaa tai nimeä klikkaamalla aukeaa tietosivu, jossa on monipuolisesti tietoa ja lisäkuvia. Myös lähteet on merkitty, mikäli tietoa on haettu muualta. Varsinaisia hoito-ohjeita ei anneta, mutta etenkin luonnossa esiintyvistä kasveista löytyy tietoa erittäin monipuolisesti.

Suomen lajitietokeskuksen sivuilla ([laji.fi](http://laji.fi)) voi selata kasvi- ja eläinlajeja kuvineen. Putkilokasvilistalla on yli 3200 lajia. Kasvin nimeä klikkaamalla avautuu tietosivu, joka sisältää monipuolista lajitietoa. Aineisto on vapaasti käytettävissä CC-BY 4.0 -lisenssin mukaisesti. Aineisto on osittain rajoitettua lajien uhanalaisuuden tai muiden suojelullisten syiden vuoksi. Rajoitukset on yleisesti toteutettu karkeistamalla tai salaamalla sijaintitietoja. Lajitietokeskus toimii yhteistyössä iNaturalist-havaintopalvelun kanssa ja tahot keräävät yhdessä käyttäjiltä luontohavainnot ja auttavat tunnistamaan lajeja (iNaturalist Suomi, 2020).

Kaupallisten toimijoiden kasvioista maininnan arvoinen on Kekkilän kasvikirjasto, joka sisältää satoja kotipuutarhaan istutettavia kukkia ja kasveja sekä niiden hoito-ohjeita. Tietokannasta voi tehdä hakuja nimen lisäksi värin, veden tarpeen, kukinta-ajan ja valon tarpeen mukaan. Kasvin kuvaa klikkaamalla pääsee lukemaan kattavia perustietoja ja hoito-ohjeita sekä vinkkejä.

PlantNet tarjoaa sovelluksen lisäksi rajapinnan heidän kuvantunnistusohjelmaansa, jolla tietokannan sisältöä voi tarkastella ja lisätä kasvitietoja tietokantaan. Rajapinnan kautta voi tehdä 50 kyselyä päivässä ilman veloitusta, joten ohjelmointitaitoinen harrastelija voi hyvinkin kiinnostua



tästä tietokannasta. Rajapintakuvaukset löytyvät osoitteesta <https://my.plantnet.org/>.

## 7 SOVELLUSTEN VERTAILU

Olemassa olevia kasvintunnistussovelluksia on erittäin paljon saatavilla ja teknologia niiden takana on kaikissa pitkälti samanlaista. Tässä kappaleessa vertaillaan muutamaa suurinta kasvintunnistussovellusta sekä paria puutarhasovellusta, jotka on ladattu ilmaiseksi Apple Storesta iPhone 7 puhelimeen. Vertailu on toteutettu kesällä 2020. Sovellusten vertailussa kiinnitettiin huomiota kasvin tunnistamiseen ja siihen, kuinka hyvin sovellus antaa apua aloittelijalle. Vertailua on käytetty myös rakennetun prototyypin pohjatyönä.

Google Lens -sovelluksen voi ladata Google Play kaupasta omana sovelluksenaan Androidille tai sitä voi käyttää osana muita Googlen sovelluksia. Google Lens on erittäin monipuolinen sovellus, jolla voi hakea kuvassa olevaan asiaan liittyviä tietoja Googlen hakupalvelusta. Google Lens tunnistaa kasvien ja eläinten lisäksi myös paikkoja sekä tekstiä. (Google Lens, 2020)

Google Lens vertailee kuvan kohdetta muihin hakupalvelun kuviin ja arvioi kuvien samankaltaisuutta sekä relevanssia alkuperäistä kuvaa vasten. Sovellus käyttää lisäksi ymmärrystään kasveista sekä kuvien metadattaa antaakseen vaihtoehtoja hakutulokselle. (Google Lens, 2020)

Hakutulokset näkyvät myös suomen kielellä, mikäli käyttäjällä on kielivaihtokielinä suomi Googlen haussa. Virallisesti Google Lens -sovellusta ei saa vielä suomenkielisenä. Sovellus on maksuton. Aloittelijalle sovellus sopii erinomaisesti, sillä tulokset ovat käytännössä Google-hakuja eli tietoa on saatavilla todella paljon.

iNaturalist ja PlantNet ovat kansalaisprojekteja, joiden toiminta perustuu käyttäjien raportoimiin havaintoihin ja kommentointiin. iNaturalistin taustalla ovat California Academy of Sciences ja National Geographic society (iNaturalist, 2020).

Lähtökohta näille sovelluksille on tieteellinen ja Suomen Lajitietokeskus on mukana kehittämässä iNaturalistin paikallistuntemusta. Käyttäjien havaintoja myös kuratoidaan, mikä parantaa projektin uskottavuutta. PlantNetin tietokannassa olevia kuvia käytetään PlantCLEF-haasteessa, jonka tarkoituksena on kehittää muun muassa konenäköön ja hahmontunnistukseen liittyvää teknologiaa. (ImageCLEF, 2020).

iNaturalist hankkeeseen voi osallistua kuka vain iNaturalistin nettisivuilla tai iOS ja Android -sovelluksia käyttämällä. iOS-sovelluksesta ei toisiksi ole saatavilla suomenkielistä versiota. PlantNetin sovellus toimii myös suomeksi. Aloittelevalle puutarhurille sovellukset kyllä kertovat, mikä laji on kyseessä, mutta varsinaisia ohjeita niiden hoitoon tai muuhun

käsittelyyn ei tarjota. PlantNetissä joutuu myös itse valitsemaan vastauksen annetuista vaihtoehdoista tai odottamaan muiden käyttäjien kommentteja.

PlantSnapin mukaan (PlantSnap, 2019) sovelluksen tietokannassa on yli 600 000 tuhatta kasvia. Sovelluksen käyttö perustuu käyttäjien raportointiin havaintoihin, kuten iNaturalistin ja PlantNetinkin. Sovelluksessa on kartta, josta näkee missä päin käyttäjät ovat tehneet havaintoja. Tämä on kuitenkin maksullinen ominaisuus ja ilmaisversiossa kartalla näkyvät vain käyttäjän omat lisäykset. Ilmaisversiossa voi tunnistaa enintään 10 kuvaa päivässä. Laajan tietokannan vuoksi tunnistuksen saa heti eikä käyttäjän tarvitse odottaa muiden käyttäjien kommentteja enää.

PlantSnap on saatavilla myös suomeksi, mutta hakutulokset kertovat ainoastaan kasvin tieteellisen ja englanninkielisen nimen. Lisäksi annetaan kuvaus kasvista sekä perustietoja. Sovelluksen kautta voi nähdä myös ostopaikkoja, mistä kyseisen kasvin voi tilata, mutta nämä ovat pääosin ulkomaisia verkkokauppoja.

Puutarhanhoitoon keskittyvistä sovelluksista vertailuun valikoituivat Blossom, myGarden ja Gardenia. Blossom tunnistaa kasveja kuvasta ja tarjoaa myös hoito-ohjeita. Sovellukseen voi asettaa itselleen muistutuksia, joista saa halutessaan push-viestin. Sovelluksen tietokannasta löytyviä kasveja voi lisätä omaan kokoelmaansa, ja niille voi lisätä muistutuksia liittyen kasteluun, lannoitukseen ja uuteen ruukkuun siirtämiseksi. Sovellus on erittäin helppokäyttöinen ja selkeä. Sovelluksesta voi lukea myös hyödyllisiä vinkkejä ja muita puutarhan hoitoon liittyviä artikkeleita. Sovellus sopii erittäin hyvin myös aloittelijalle. Suomenkielisenä sitä ei ole tarjolla.

myGarden ei tunnista kasveja, mutta sovelluksen tietokannasta voi etsiä kasveja nimellä ja lisätä ne omiin tietoihinsa. Sovellukseen voi lisätä itselleen tehtäviä, mutta niitä pitää itse muistaa käydä läpi, sillä sovellus ei muistuta käyttäjää. Lisäksi sovelluksessa on pieni määrä ohjeita ja erilaisia tutoriaaleja. myGarden-sovelluksessa voi myös suorittaa erilaisia saavutuksia lisäämällä itselleen kasveja ja suorittamalla tehtäviä. Pelillistäminen tekee sovelluksen käytöstä hieman hausempaa. Sovelluksen taustalla on puutarhavälineitä valmistava Gardena.

Gardenia-puutarhasovellus ei myöskään tunnista kasveja kuvan perusteella, mutta käyttäjä voi etsiä kasvia nimeltä sovelluksen tietokannasta ja lisätä sen omaan kokoelmaansa. Kasvin tietosivulla on selkeän minimalistisesti kerrottu kasvin menestymisedellytyksistä. Kun kasvin on lisännyt kokoelmaansa, voi sille lisätä hoitoon liittyviä muistutuksia. Muistutusten lisääminen tapahtuu isosta plusmerkistä kasvin tietosivulla. Plus-merkin takaa löytyy eri muistutusvaihtoehtoja, mutta ikonit eivät ole kovin selkeitä eikä sanallista selitystä löydy. Sovelluksen käytettävyydessä on hieman epäselvyyksiä, mikä vähentää sen houkuttelevuutta.

## 8 DESIGN SPRINT

Opinnäytetyön prototyyppiä lähdettiin rakentamaan Google Venturesin Design Sprintiä mukaillen. Design sprintin ideana on hoitaa asia viiden työpäivän aikana maanantaista perjantaihin, mutta opinnäytetyön luonteen vuoksi sprintiä ei noudatettu säännönmukaisesti. Esimerkiksi perjantainen käyttäjättestaus jätettiin opinnäytetyön laajuudesta pois ja muidenkin päivien agendoja muokattiin tilanteeseen paremmin sopivaksi.

Design sprintin ensimmäisiä tehtäviä on pohtia haasteita ja pelkoja, joita tiimillä on tavoitteen suhteen. Haasteet muutetaan kysymysmuotoon siten, että vastaukset on mahdollista selvittää sprintin aikana. Puutarhaiheisen mobiilisovelluksen osalta nousi esiin seuraavat kysymykset:

- Kuinka prototyypin saa ajoissa valmiiksi?
- Kuinka sovelluksesta tehdään mahdollisimman yksinkertaisen?
- Kuinka sovellus tunnistaa kasvit?
- Kuinka tehdään "kaunis" sovellus?
- Kuinka saadaan sovellus lopulta Apple Storeen?

Näistä ensimmäinen ja viimeinen pudotettiin pois sprintiltä, sillä ne eivät tuoneet lisäarvoa opinnäytetyöprosessiin. Design sprintin ensimmäisenä päivänä luodaan käyttöliittymän määrittäykset, kuvataan ongelma ja valitaan osa-alue, johon sprintissä keskitytään. Jotta sprintissä voi edetä seuraavaan päivään, pitää ensimmäisen päivän olla valmis. Ensimmäisen päivän tavoitteena oli selkiyttää sovelluksessa tapahtuvia käyttötappauksia.

Puutarha-sovelluksen käyttäjäpersoonaa pohjautuu fiktiiviseen henkilöön, josta ei vielä ole dataa saatavilla. Friisin ja Yu Siangin (2020) mukaan persoonaa voidaan kehittää myös design-tiimin kokemusten pohjalta. Tässä opinnäytetyössä on käytetty käyttäjäpersoonaa, joka pohjautuu kirjoittajan omiin kokemuksiin vastaavista sovelluksista sekä havaintoihin, joita on tehty Facebookin eri puutarha- ja kasviryhmissä olevien kysymysten perusteella. Puutarhasovelluksen käyttäjäpersoonaksi on Ho Tranin (2019) artikkelin pohjalta kehitelty seuraavanlainen käyttäjäpersoonaa:

35-vuotias kaupungin taajama-alueella asuva perheellinen naisoletettu, joka työskentelee toimistossa. Hän ottaa muut huomioon, on utelias ja itsenäinen, introvertti mutta erittäin kilpailuhenkinen. Hän haluaa halvalla helppohoitoisen pihan muttei tiedä entuudestaan pihan hoidosta paljoakaan. Hän ei ehdi viettää aikaa pihalla päivittäin, mutta haluaa viihtyä siellä perheensä ja ystäviensä kanssa. Hän on kiinnostunut luonnon monimuotoisuudesta, luomusta ja kasvissyönnistä. Henkilö on kärsimätön ja kiinnostuksen kohteet vaihtuvat nopeasti.

Opinnäytetyöprojektissa design sprintin toinen ja kolmas päivä sulautettiin samalle päivälle, jonka aikana piirrettiin prototyyppiin rautalankamallit

sekä tehtiin päätöksiä tulevasta prototyypistä. Kappaleessa 7 vertailtujen, olemassa olevien sovellusten pohjalta on valittu prototyypissä käytettävä neliosainen päävalikko sekä tärkeimmät toiminnallisuudet, kuten kasvien tunnistus kuvasta ja muistutukset. Tehdyistä luonnoksista valittiin lopulta parhaat versiot, joiden pohjalta prototyyppiä lähdettiin tekemään.

Neljäs päivä oli prototyypin luontipäivä, jolloin tiimiläisille jaetaan roolit. Opinnäytetyöprojektissa tähän vaiheeseen meni useita päiviä, sillä opinnäytetyön tekijä toimi kaikissa rooleissa. Aluksi käytettiin Marvel-suunnitteluoohjelmaa, joka toimii ilmaiseksi selaimessa. Ohjelmalla on melko helppoa tuottaa rautalankamalleja ja toimiva prototyyppi. Lopullinen esiteltävä prototyyppi on kuitenkin luotu Figma-suunnittelutyökalulla.

Sovelluksen ulkoasuksi valittiin vaaleaa vihreää ja vaaleanpunaista. HEX-koodit käytetyille sävyille on lueteltu alla ja kuvassa 1 esitellään väripaletti.

- #F4F9F4 – Vaalea pääväri
- #A7D7C5 – Vihreä väri napeille ja reunoille
- #74B49B – Tummempi vihreä tehosteväriä
- #5C8D89 – Tumma vihreä tekstiväriä



*Kuva 1. Sovelluksen väripaletti.*

Sovelluksen taustakuvana on käytetty Unsplash-kuvanjakopalvelusta ladattua kirsikkapuukuvaa, jonka on ottanut nimimerkki Hunt Han.

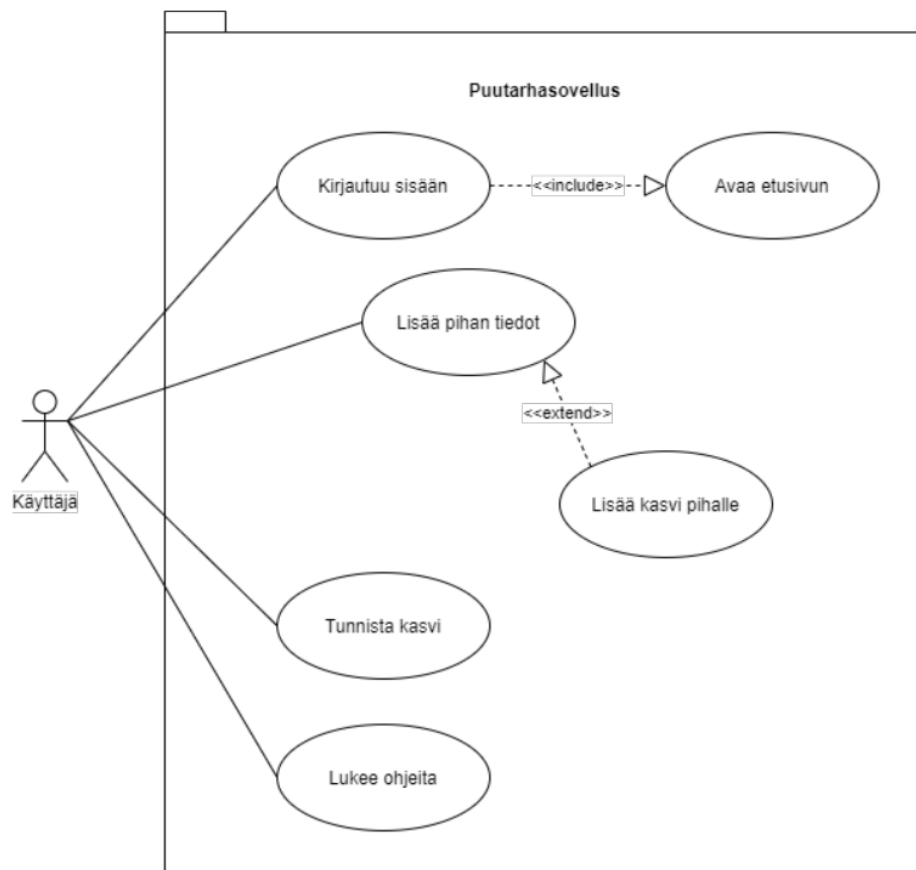
## 9 SOVELLUKSEN MÄÄRITYSDOKUMENTAATIO

Tämä kappale esittelee puutarhasovelluksen määrittäydokumentaaion. Dokumentaaionssa kuvataan puutarhasovelluksen toiminnalliset vaatimukset. Dokumentin tarkoituks on kertoa, kuinka ohjeistavan puutarhanhoitosovelluksen tulisi toimia ja ohjata sovelluksen suunnittelua ja kehitystyötä. Dokumentissa kuvataan puutarhasovelluksen vähimmäisvaatimukset, jotka tukevat osaltaan prototyypin rakentamista.

Puutarhasovellus kehitetään natiivisovellukseksi, joka toimii sekä iOS että Android-laitteissa. Sovellus neuvoo ja antaa vinkkejä aloitteleville puutarhureille sekä kaikille aiheesta kiinnostuneille. Sovelluksen käyttökielenä on suomi tai vaihtoehtoisesti englanti. Sovellusta voi käyttää mobiililaitteilla sekä vaak- että pystysuorassa.

Puutarhasovelluksen tarkoituksena on auttaa hoitamaan pihaa ympäri vuoden, tutkia olemassa olevaa kasvillisuutta sekä antaa tukea pihan hoitajalle eri toimenpiteissä. Käyttäjä pystyy tunnistamaan kasveja sekä vastaanottamaan viestejä sovellukselta, kun hän on kirjautuneena sovelluksen käyttäjäksi.

Sovelluksen kohderyhmää ovat kaupunkialueella asuvat puutarhaharrastelijat, joilla on vähän tai ei ollenkaan kokemusta pihan hoidosta, ja jotka ovat tottuneet käyttämään mobiilisovelluksia. Käyttäjäpersoonaa esitellään tarkemmin kappaleessa 8. Prototyyppi on esitelty kappaleessa 10. Puutarhasovelluksen käyttötapaukset mietittiin design sprintin ensimmäisen päivän aikana. Käyttötapaukset on kuvattu myös UML-kaaviona kuvassa 2. Tekstimuotoiset käyttötapaukset ovat liitteessä 1.



Kuva 2. Puutarhasovelluksen käyttötapaukset.

Ei-toiminnalliset vaatimukset on kuvattu liitteessä 2. Toiminnalliset vaatimukset noudattelevat käyttötapauksia, joten niitä ei ole erikseen kirjattu tähän dokumenttiin. Käyttötapausten avulla voidaan myöhemmin luoda tarkemmat toiminnalliset vaatimukset tai user storyt.

## 10 PROTOTYYPIN ESITTELY

Sovelluksen prototyyppi on luotu suunnitteluohjelma Figmalla. Prototyyppiä voi kokeilla osoitteessa <https://bit.ly/31hp3pj>. Prototyyppiä tul-  
laan säilyttämään siihen asti, kun se koetaan sovelluksen kehityksen kan-  
nalta tarpeelliseksi. Prototyypin kehitys ja muotoilu myös jatkuu vielä  
opinnäytetyöprosessin jälkeen, joten linkin takaa löytyvä proto voi erota  
tässä kappaleessa esitellyistä kuvista.

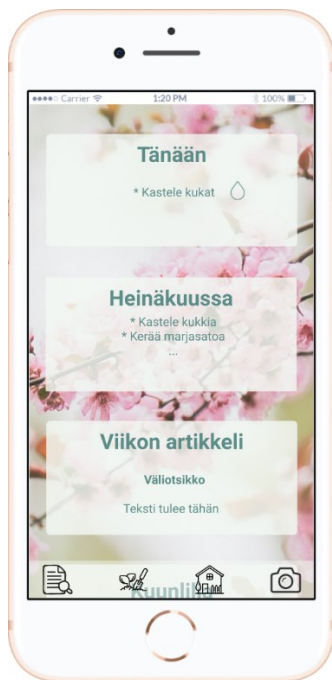
Kirjautumisprosessiin on luotu kaksi ruutua. Käyttäjä voi luoda oman tun-  
nuksen, jolla kirjautuu sovellukseen tai hän voi kirjautua Facebookin tai  
Googlen tunnuksillaan. Tämä on yleinen käytäntö kirjautumista vaativilla  
sovelluksilla, sillä ihmiset keskeyttävät helposti sovelluksen käytön, mikäli  
kirjautumisprosessi on raskas. Prototyypissä oletuksena on, että käyttäjä  
on jo aiemmin rekisteröinyt sähköpostiosoitteensa ja on nyt tullut kirjaui-  
tumaan sisään. Kuvassa 3 on esitelty Kirjautuminen sekä Salasanan tilaus  
sivut.



Kuva 3. Kirjautumiseen liittyvät näkymät.

Kirjautumisen jälkeen avautuvalla etusivulla on muistutuksia, ohjeita ja  
artikkeleita. Etusivu näkyy kuvassa 4. Ruudun alaosassa on päävalikko,  
josta pääsee siirtymään osiosta toiseen miltä sivulta tahansa. Artikkelit,  
jotka avautuvat etusivulta tai muokattavat sivut voi sulkea X-ikonista, jol-  
loin käyttäjä palaa näkymään, josta klikkasi auki kyseisen sivun.





Kuva 4. Etusivu ja uusimmat artikkelit.

Ruutua pystyy vierittämään vertikaalisesti, mikäli tietoa on enemmän kuin näkyviin mahtuu. Statusvalikko sekä sovelluksen päävalikko ovat näkyvillä koko ajan. Päävalikossa etusivu on merkitty ikonilla, jossa on suurenruskasi ja tekstisivu. Etusivu jakautuu neljään osioon:

**Tänään** – Osiossa näkyvät päivän muistutukset. Klikkaamalla pääsee hoito-ohjeet-sivulle. Muistutukset määräytyvät automaattisesti valitun hoito-ohjelman mukaisesti tai niitä voi käyttäjä itse lisätä.

**Tässä kuussa** – Kuukauden tehtävälista, johon on lisätty yleisiä hoito-ohjeita ja vinkkejä kyseiselle kuukaudelle.

**Viikon artikkeli** – Viikottain vaihtuva artikkeli. Yleisiä puutarhanhoitoon liittyviä juttuja. Sovellus voi kerätä näitä eri piha- ja puutarhasivustoilta ja linkittää juttuihin, jolloin artikkeleita ei tarvitse erikseen keksiä vain tätä sovellusta varten.

**Kuukauden kasvi** – Joka kuukausi esitellään kasvi, jolle annetaan tarkemmat hoito-ohjeet. Kasvi voi olla myös luonnonvarainen, jolloin kerrotaan, kuinka kasvi huomioidaan. Onko se esimerkiksi rauhoitettu tai hävitettävä vieraslaji. Esimerkissä esitellään Kuunlija-niminen perennakasvi.

Kuvassa 5 näkyvän viikon artikkelin saa suljettua vasemman yläkulman X-ikonista ja käyttäjä palautuu etusivulle.



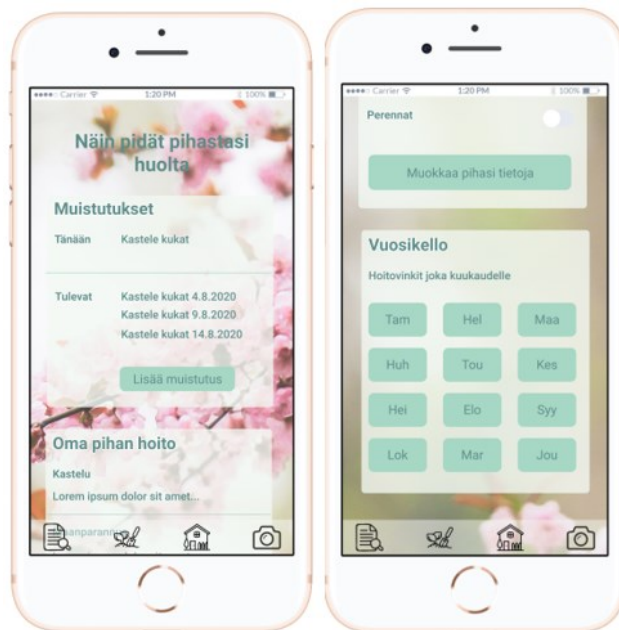
Kuva 5. Artikkelinäkömä.

Hoito-ohjeet-osio löytyy päävalikon ikonista, jossa on hara ja maasta nousvat lehdet. Osiossa on useita vaihtoehtoja, joihin käyttäjä voi keskittyä. Jos käyttäjä on lisännyt oman pihansa tiedot sovellukseen, sovellus antaa pihan tietojen perusteella ohjeita kasteluun ja esimerkiksi lannoitukseen tai maanparannukseen liittyen. Sovellus muistuttaa toimenpiteistä push-ilmoituksella ja etusivun Tänään-osiossa, kun se on ajankohtaista.

Muistutuksia voi asettaa myös itse. Uusi muistutus -sivu aukeaa omalle ruudulle ja sivulla on pieni lomake, johon lisätään aihe, ajankohta ja mahdollinen toisto. Ajankohta ja toisto valitaan, kuten älypuhelimien kalentrisovelluksissa.

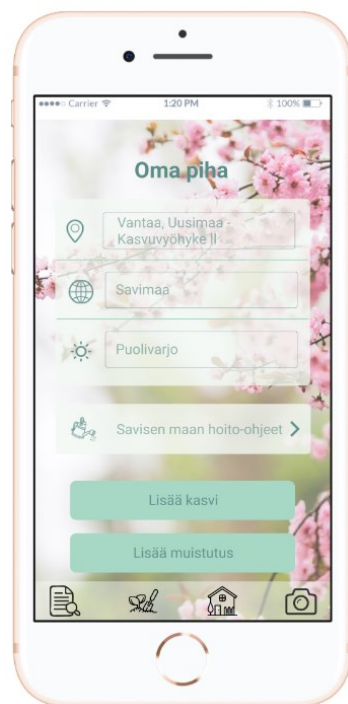
Oman pihan hoito-ohjeita voi halutessaan muokata, vaihtoehtoja on prototyypissä muutama. Käyttäjällä pystyy valitsemaan hoito-ohjelman kytkimellä päälle tai pois päältä.

Hoito-ohjeet osiossa on myös pihan hoidon vuosikello, josta voi kuukausikohtaisesti nähdä ohjeita ja vinkkejä. Kuukausikohtaiset tehtävät avautuvat uuteen ruutuun, jonka saa suljettua X-ikonista. Vuosikello ja muistutukset näkyvät kuvassa 6.



Kuva 6. Näkymiä hoito-ohjevalikosta.

Kuvassa 7 näkyvässä Oma piha -näkyvässä käyttäjä voi lisätä oman pihansa tietoja sovellukseen. Näkymä löytyy päänavigaation talokuvakkeen takaa. Jos sijaintitiedot on jaettu sovellukselle, sovellus hakee sijainnin automaattisesti ja kertoo vastaavan kasvuvyöhykkeen. Käyttäjä voi halutessaan myös kirjoittaa paikkakunnan, jolloin sovellus ennakoii ja antaa kirjainten perusteella vaihtoehtoja, jotta käyttäjän ei tarvitse kirjoittaa kokonaan kaupungin nimeä.

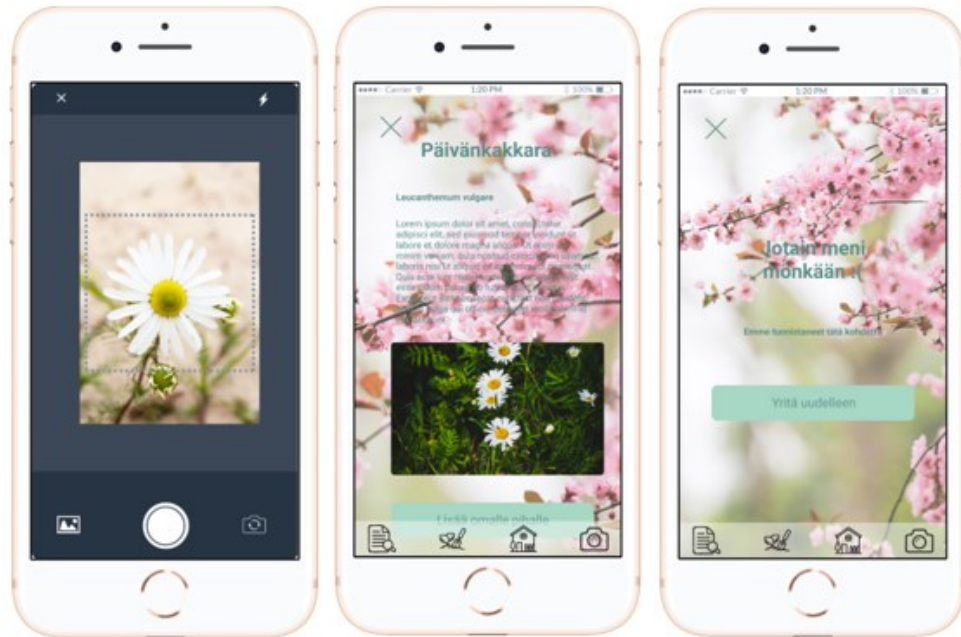


Kuva 7. Oma piha -näkyvä.

Maaperän voi valita maapallokuvakkeen vieressä olevasta vetovalikosta, jossa on eri vaihtoehtoja. Valo-olosuhteet voi myös kirjata oman pihan tietoihin, jolloin sovellus osaa ehdottaa pihalla menestyviä kasveja.

Oma piha -osiossa voi myös lisätä kasveja pihalleen. Lisää kasvi -napista avautuu kameranäkymä, jolloin käyttäjä voi ottaa kuvan tai valita sen galleriastaan. Kasvitietosivulla Lisää omalle pihalle -napista kasvi lisätään omat kasvit osioon, joka näkyy vain, jos kasveja on.

Tunnistusosiossa käyttäjä ottaa kuvan näkemästään kasvista tai käyttää laitteensa kuvakirjaston kuvaa. Sovellus näyttää tunnistetusta kasvista perustiedot ja hoito-ohjeet. Sovellus myös kertoo, mikäli kasvia ei voida tunnistaa. Kuvassa 8 näkyvät tunnistukseen liittyvät näkymät.



Kuva 8. Kuvan otto, kasvisivu ja virheilmoitus.

## 11 YHTEENVETO

Opinnäytetyön tavoitteena oli rakentaa mobiilisovelluksen käyttöliittymä-prototyyppi sekä luoda alustava vaatimusmäärittäydokumentaatio mahdollista jatkokehitystä varten. Prototyypin rakentamisessa käytettiin Design sprint -menetelmää, joka auttoi hyvin huomioimaan monia suunnitteluun ja käytettävyyteen liittyviä asioita. Osittain prototyyppiin jäi edelleen korjattavia asioita ja etenkin ruudun alalaidassa kelluva päänavigaatio saattaa vaatia jatkokehitystä. Seuraavaksi olisi hyvä antaa prototyyppi muutaman käyttäjän testattavaksi ja korjata käytettävyyttä testien perusteella. Prototyypin osalta tavoite saavutettiin.

Sovelluksen määrittäydokumentaatiossa pyrittiin selkeästi esittelemään, miten sovelluksen pitää toimia. Dokumentaatiossa kuvataan tärkeimmät käyttötapaukset sekä ei-toiminnallisia vaatimuksia, joita jatkokehityksessä tarvitaan. Vaatimukset on pyritty pitämään yksinkertaisina, vaikka toisinaan taustalla tapahtuukin paljon asioita käyttäjän sitä huomauttamatta.

Kasvien tunnistaminen kuvasta osoittautui melko haastavaksi tutkimuskohteeksi. Opinnäytetyössä tutkittiin kahta eri tekniikkaa käyttäviä sovelluksia. Vanhemmat sovellukset pohjaavat tekoälynä konenäköön ja ohjattuun oppimiseen. Uudemmat sovellukset puolestaan käyttävät syväoppivia neuroverkkoja, jotka oppivat itseksensä ja ovat erittäin tehokkaita. Tämän osion kanssa jouduin hieman kamppailemaan, koska aihe sisältää paljon käsitteitä ja prosesseja, jotka eivät olleet itselle kovin tuttuja entuudestaan. Asiat on pyritty kuvaamaan siltä osin kuin ne ovat suunnitteluprosessin kannalta olennaisia.

Sovellussuunnittelun näkökulmasta kasvitietokantojen tutkiminen jäi hieman triviaaliksi mutta pääsin tutustumaan moneen mielenkiintoiseen portaaliin tutkimuksen myötä. Tietokantoja tutkiessa selvisi myös PlantNetin tarjoama API, jota voisi kokeilla omassakin puutarhasovelluksessa tiettyyn pisteeseen asti. Opinnäytetyöprojekti oli erittäin mielenkiintoinen ja opettavainen. Sovelluksen kehitys jatkuu edelleen harrasteprojektina.

## LÄHTEET

- Affouard, A. & Goeau, H. & Bonnet, P. & Lombardo, J. & Joly, A. (2017) Pl@ntnet app in the era of deep learning. Haettu 20.8.2020 osoitteesta <https://openreview.net/pdf?id=HJVJpENFg>
- Agile Alliance. (2020.) What is Agile Software Development? Haettu 13.8.2020 osoitteesta <https://www.agilealliance.org/agile101/>
- Alvertis, I. & Koussouris, S. & Papaspyros, D. & Arvanitakis, E. & Mouzakis, S. & Franken, S. & Kolvenbach, S. & Prinz, W. (2016) User Involvement in Software Development Processes. *Procedia Computer Science* Volume 97, 2016. ss. 73-83. Haettu 14.8.2020 osoitteesta <https://doi.org/10.1016/j.procs.2016.08.282>
- Brownlee, J. (2019) A Gentle Introduction to Computer Vision. Päivitetty 5.7.2019. Haettu 10.8.2020 osoitteesta <https://machinelearningmastery.com/what-is-computer-vision/>
- Budiu, R. (2019). Mobile User Experience: Limitations and Strengths. Haettu 5.8.2020 osoitteesta <https://www.nngroup.com/articles/mobile-ux/>
- Friis Dam, R & Yu Siang, T. (2020). Personas – A Simple Introduction. Interaction Design Foundation. Haettu 4.8.2020 osoitteesta [www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them](http://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them)
- GeeksforGeeks. (2018.) Software Engineering | Classification of Software Requirements. Haettu 13.8.2020 osoitteesta [www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/](http://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/)
- Google Lens. (2020). What is Google Lens? Haettu 23.6.2020 osoitteesta [lens.google.com/howlensworks/](https://lens.google.com/howlensworks/)
- GV. (2019). The Design Sprint. Haettu 10.7.2020 osoitteesta <https://www.gv.com/sprint/>
- GV Library. (2016). <https://library.gv.com/sprint-week-tuesday-d22b30f905c3>
- Ho Tran, T. (2019). 5 essentials for your user persona template (with examples). Inside Design Blog. Blogijulkaisu 30.9.2019. Haettu 4.8.2020 osoitteesta [www.invisionapp.com/inside-design/user-persona-template/](http://www.invisionapp.com/inside-design/user-persona-template/)
- ImageCLEF. (2020.) ImageCLEF - The CLEF Cross Language Image Retrieval Track. Haettu 23.06.2020 osoitteesta <https://www.imageclef.org/>

iNaturalist. (2020). What is iNaturalist? Haettu 23.06.2020 osoitteesta <https://www.inaturalist.org/pages/help#general1>

iNaturalist Suomi. (2020). iNaturalist Suomi. Haettu 18.8.2020 osoitteesta <https://inaturalist.laji.fi/pages/about-suomi>

Interaction design foundation. (n.d.). Heuristic evaluation. Haettu 17.08.2020 osoitteesta <https://www.interaction-design.org/literature/topics/heuristic-evaluation>

Kumar, N. & Belhumeur, P. & Biswas, A. & Jacobs, D. & Kress, J. & Lopez, I. & Soares, J. (2012.) Leafsnap: A Computer Vision System for Automatic Plant Species Identification. (Oral presentation). *Proceedings of the 12th European Conference on Computer Vision (ECCV)*. Haettu 10.8.2020 osoitteesta [https://neerajkumar.org/projects/leafsnap/base/papers/nk\\_eccv2012\\_leafsnap.pdf](https://neerajkumar.org/projects/leafsnap/base/papers/nk_eccv2012_leafsnap.pdf)

Laakso, S. (2006). Käyttöliittymälähtöinen vaatimusmäärittely. *Systemityö 2/2006*. ss: 19-21. Haettu 13.8.2020 osoitteesta [www.sytyke.org/wp-content/uploads/2017/06/Systemity%C3%B6-lehti\\_2-2006.pdf](http://www.sytyke.org/wp-content/uploads/2017/06/Systemity%C3%B6-lehti_2-2006.pdf)

Leppäniemi, P. (2012). Ohjelmistojen määrittely, mallintaminen ja dokumentointi. Haettu 13.08.2020 osoitteesta <http://ohjelmistotuotanto.panuleppaniemi.com/maarittely-mallintaminen-dokumentaatio/>

Meom. (2020). Design Sprint. Haettu 10.7.2020 osoitteesta <https://www.meom.fi/design-sprint/>

MobileSyrup. (2018.) Identify plants with an artificial intelligence app [App of the Week]. Haettu 10.8.2020 osoitteesta <https://mobilesyrup.com/2018/03/10/identify-plants-artificial-intelligence-app-plantsnap/>

Nielsen, J. (2011). Mobile UX Sharpens Usability Guidelines. Haettu 5.8.2020 osoitteesta <https://www.nngroup.com/articles/mobile-sharpens-usability-guidelines/>

Nielsen, J. (2012). Usability 101: Introduction to Usability. Haettu 18.7.2020 osoitteesta <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

Phys.org. (2011.) Professor develops mobile app to identify plant species. Päivitetty 8.6.2011. Haettu 10.8.2020 osoitteesta <https://phys.org/news/2011-06-professor-mobile-app-species.html>

PlantSnap. (2019). What sets us apart - instantaneous results. Haettu 23.06.2020 osoitteesta [www.plantsnap.com/who-we-are/](http://www.plantsnap.com/who-we-are/)

Reaktor. Neuroverkkojen periaatteet. Elements of AI verkkokurssin aiheisto. Haettu 10.8.2020 osoitteesta <https://course.elementsofai.com/fi/5/1>

Schwaber, K. & Sutherland, J. (2017). The Scrum Guide™ The Definitive Guide to Scrum: The Rules of the Game. Haettu 13.08.2020 osoitteesta <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>

Service Design Network. (2020.) What is Service Design? Haettu 14.8.2020 osoitteesta [www.service-design-network.org/about-service-design](http://www.service-design-network.org/about-service-design)

Shneiderman, B. (N.d.). The Eight Golden Rules of Interface Design. Haettu 9.8.2020 osoitteesta <https://www.cs.umd.edu/users/ben/goldenrules.html>

Stellman, A. & Greene, J. (2005) *Applied Software Project Management*. Sebastopol, CA, USA: O'Reilly Media, Inc.

Tractinsky, N. (2020). Visual Aesthetics. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* Haettu 6.8.2020 osoitteesta [www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/visual-aesthetics](http://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/visual-aesthetics)

Usability.gov. (2020). Use Cases. Haettu 23.7.2020 osoitteesta <https://www.usability.gov/how-to-and-tools/methods/use-cases.html>



## KÄYTTÖTAPAUKSET

<b>ID</b>	UC 1
<b>Nimi ja versio</b>	Käyttäjä kirjautuu sovellukseen v1
<b>Suorittajat</b>	Normaali käyttäjä
<b>Esiehdot</b>	Käyttäjä on ladannut sovelluksen sovelluskaupasta ja rekisteröitynyt käyttäjäksi
<b>Kuvaus</b>	Käyttäjä kirjautuu sovellukseen. Vaihtoehtoina Facebook, Google tai AppleID:llä kirjautuminen sekä oman tunnuksen luonti sähköpostia ja salasanaa käyttäen.
<b>Poikkeukset</b>	Sähköpostin vahvistaminen rekisteröitymisen yhteydessä.
<b>Lopputulos</b>	Kirjautumisen jälkeen sovellus aukeaa suoraan etusivulle.
<b>Muut vaatimukset</b>	n/a

<b>ID</b>	UC 2
<b>Nimi ja versio</b>	Käyttäjä lisää pihansa tiedot sovellukseen v2
<b>Suorittajat</b>	Normaali käyttäjä
<b>Esiehdot</b>	Käyttäjä on kirjautuneena sovellukseen
<b>Kuvaus</b>	Käyttäjä lisää oman pihan tiedot sovelluksen oma piha osiossa. Käyttäjä valitsee pihaan liittyviä ominaisuuksia saadakseen ajan-kohtaiset ja oikeat hoito-ohjeet.
<b>Poikkeukset</b>	
<b>Lopputulos</b>	Pihan tiedot lisätään sovellukseen. Pihan hoito-ohjelma muodostuu.
<b>Muut vaatimukset</b>	Käyttäjä voi antaa sovellukselle sijaintitietonsa pihan sijaintipaikkakuntaa varten. Pihan tiedot voi poistaa.

<b>ID</b>	UC 3
<b>Nimi ja versio</b>	Käyttäjä lisää kasvin pihallensa v2
<b>Suorittajat</b>	Normaali käyttäjä
<b>Esiehdot</b>	Käyttäjä on kirjautuneena sovellukseen
<b>Kuvaus</b>	1. Käyttäjä tunnistaa kasvin 2. Käyttäjä lisää kasvin omaan pihaansa.
<b>Poikkeukset</b>	Sovellus ei voi lisätä kasvia pihalle
<b>Lopputulos</b>	Kasvi näkyy Oma piha -osiossa Hoito-ohjeet päivittyvät
<b>Muut vaatimukset</b>	Kasveja voi lisätä Oma piha -osioon, vaikka muita tietoja ei ole vielä lisätty.

<b>ID</b>	UC 4
<b>Nimi ja versio</b>	Käyttäjä tunnistaa kasveja sovelluksella
<b>Suorittajat</b>	Normaali käyttäjä
<b>Esiehdot</b>	Käyttäjä on kirjautuneena sovellukseen. Sovellukselle on annettu oikeudet puhelimen kameran ja kuvakirjaston käyttöön.
<b>Kuvaus</b>	1. Käyttäjä ottaa kuvan kasvista tai lataa kuvan kuvakirjastosta 2. Sovellus hakee kasvin tiedot tietokannasta 3. Sovellus näyttää kasvin tiedot ja hoito-ohjeet.
<b>Poikkeukset</b>	Sovellus ei tunnista kasvia
<b>Lopputulos</b>	Sovellus hakee kasvin tiedot ja kertoo hoito-ohjeet.
<b>Muut vaatimukset</b>	Haku kestää korkeintaan 5 sekuntia. Jos kasvia ei löydy, annetaan vaihtoehtona lähin arvaus tai virheilmoitus

<b>ID</b>	UC 5
<b>Nimi ja versio</b>	Käyttäjä vastaanottaa pihan hoito-ohjeita
<b>Suorittajat</b>	Normaali käyttäjä
<b>Esiehdot</b>	Käyttäjä on kirjautuneena sovellukseen. Push-viestit on sallittu.
<b>Kuvaus</b>	Käyttäjä saa ajantasaiset hoito-ohjeet sovellukseen sekä halutessaan push-viestinä puhelimeensa.
<b>Poikkeukset</b>	-
<b>Lopputulos</b>	Puutarhan ajankohtaiset hoito-ohjeet ovat käyttäjän saatavilla
<b>Muut vaatimukset</b>	N/A

## EI-TOIMINNALLISET VAATIMUKSET

ID	Nimi	Vaatimuksen kuvaus	Perustelut	Luokka	Prioriteetti
NF4	iOS ja Android tuki	Sovellus toimii sekä iOS että Android järjestelmissä ja noudattelee käyttöjärjestelmien design ja styleguideja.	Käytettävyys, Saatavuus	Ei-toiminnallinen	1 - Kriittinen
NF5	Sovellus on käytettävissä 24/7	Sovellus on käytettävissä yli 99 % ajasta.	Saatavuus; Sovelluksen käyttö ei ole aikarajoitettua vaan sitä saatetaan käyttää myös epätavallisina aikoina sekä eri aikavyöhykkeellä.	Ei-toiminnallinen	1 - Kriittinen
NF6	Noudattaa OWASP standardia	Sovelluksessa 0 kappaletta haavoituvuuksia, jotka on listattu OWASP Mobile Application Security Checklistillä.	Tietoturva	Ei-toiminnallinen	1 - Kriittinen
NF7	Tietojen säilytys	Käyttäjätiedot tallennetaan turvallisesti. Jos käyttäjä ei avaa sovellusta vuoteen, käyttäjätiedot poistetaan.	Tietoturva, GDPR	Ei-toiminnallinen	1 - Kriittinen
NF2	Kasvihaun kesto	Kuvan ottamisen tai valitsemisen jälkeen, sovellus etsii vastauksia maksimissaan 5 sekuntia.	Käytettävyys, Suorituskyky	Ei-toiminnallinen	2 - Korkea

ID	Nimi	Vaatimuksen kuvaus	Perustelut	Luokka	Prioriteetti
NF8	Käyttäjän täppäykset ja swai- pit huomioidaan heti	Kun käyttäjä näpäyttää element- tiä, toiminto aktivoituu heti ja siir- tymä seuraavaan näkymään ta- pahtuu alle 0,5 sekunnissa.	Käytettävyys	Ei-toiminnallinen	2 - Korkea
NF1	Suomenkielinen käyttöliittymä	Sovellusta on mahdollista käyttää suomen kielellä.	Käytettävyys	Ei-toiminnallinen	3 - Matala
NF3	Käyttöliittymä toimii eri asunnoissa	Sovellus toimii mobiililaitteilla pysty- ja vaakänäkyvässä.	Käytettävyys	Ei-toiminnallinen	3 - Matala