ÄLYPALLON OHJELMISTOKEHITYSPROJEKTI

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tietotekniikan kandidaatintyö

2024

Samuel Repo

Tarkastaja: Tohtori Sonja Hyrynsalmi

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tietotekniikka

Samuel Repo

Älypallon ohjelmistokehitysprojekti

Tietotekniikan kandidaatintyö

2024

28 sivua, 5 kuvaa ja 1 taulukko

Tarkastaja: Tohtori Sonja Hyrynsalmi

Avainsanat: Projektityö, kehityskaari, ohjelmistokehitys, MVP-taso

Työssä kuvataan Älypallo -nimiselle hierontalaitteelle kehitetyn ohjainsovelluksen kehitysprosessia. Työtä taustoitetaan kertomalla Älypallosta, siinä käytetyistä teknologioista, ohjainsovelluksen tarpeellisuudesta ja työssä käytettävistä periaatteista ja tekniikoista. Kehitysprosessia kuvataan vaatimusmäärittelyn, suunnittelun, toteutuksen ja testauksen kannalta.

Vaatimusmäärittely esitellään kokonaisuudessaan ja siitä käytyä keskustelua avataan, pääosin kertomalla mitä osia vaatimusmäärittelystä päätettiin hylätä. Suunnittelutyötä avataan UML-mallinnusten ja kaavioiden, esimerkiksi käyttötapauskaavion, esittelyn kautta. Toteutuksesta ja testauksesta annetaan konkreettisia esimerkkejä, kehitystyöstä kerrotaan menemättä kuitenkaan kovin sisälle itse kirjoitettuun koodiin. Testausta avataan yksittäisten testitapausten kautta.

Lopputulemana esitellään sovelluksen kehityksen koko kaarta, ja lopuksi kehitystyön onnistumisesta käydään keskustelua ja pohditaan mahdollisia jatkokehityksen kohteita sovellukselle.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Software Engineering

Samuel Repo

Smartball software development project

Bachelor's thesis

2024

28 pages, 5 figures and 1 table

Examiners: PhD Sonja Hyrynsalmi

Keywords: Project, development cycle, software development, MVP-level

The thesis describes the development process of a control application for a massage device called Smartball. The background to the work is given by explaining the Smartball, the technologies used in it, the need for the controller application and the principles and techniques used in the work. The development process is described in terms of requirements definition, planning, development and testing.

The specification is presented in full, and the discussion of the specification is opened, mainly by explaining which parts of the specification it was decided to discard. The planning work is opened through the presentation of UML modelling and diagrams, such as the use case diagram. Concrete examples of development and testing are given, and the development work is described without going too deeply into the code that was written. Testing is opened through individual test cases.

As a final result, the whole development cycle of the application is presented, and finally, the success of the development is discussed and possible areas for further development of the application are considered.

KIITOKSET/ ACKNOWLEDGEMENTS

Sonja Hyrynsalmi kannustamisesta ja tukemisesta

LYHENNELUETTELO

Lyhenteet

MVP	pienin toimiva tuote (minimum viable product)
NIR	lähi-infrapunasäteily (near-infrared)
UML	koottu mallinnuskieli (unified modeling language)
SQL	kyselykieli (structured query language)

Symbolit

nm	nanometri
THz	terahertsi

Sisällysluettelo

Tiivistelmä

Abstract

Kiitokset

Lyhenneluettelo

1 Johdanto 10
1.1 Taustaa 10
1.2 Älypallon fyysinen rakenne 11
2 Työn rakenne 12
2.1 Vaatimusmäärittely 12
2.2 Suunnittelu 13
2.3 Kehitys 13
2.4 Testaus 14
3 Projektin toteutus 16
3.1 Vaatimusmäärittelyn muutokset 16
3.1.1 Vaatimusmäärittelyn sisältö 16
3.1.2 Keskustelu vaatimusmäärittelystä 18
3.2 Suunnitelmat 18
3.2.1 Paperiset käyttöliittymämallit 19
3.2.2 Toimintokaaviot 20
3.3 Kehitystyö 22
3.3.1 Käyttöliittymä 22
3.3.2 Toiminnallisuus 24
3.4 Testaaminen 24
4 Lopputulema 26
Lähteet 27

Kuvaluettelo

Kuva 1: Vaatimusmäärittelydokumentti

Kuva 2: Käyttöliittymäsuunnitelma

Kuva 3: Sekvenssikaavio

Kuva 4: Hoitohistoria

Kuva 5: Hoitosivu

Taulukkoluettelo

Taulukko 1: Käyttötapaukset

1 Johdanto

Työn aiheena on raportoida Älypallo -nimisen hieromalaitteen ohjainsovelluksen ohjelmistokehitysprosessista. Työ käsittelee kokonaisuudessaan ohjelmiston vaatimusmäärittelyn, suunnittelun, kehityksen ja testauksen. Ohjelmistosta kehitetään tämän projektin aikana MVP (minimum viable product) -tasoinen versio, jota on tarkoitus esitellä itse fyysisen laitteen yhteydessä mahdollisille sijoittajille. Laitteen toimintaperiaate perustuu NIR-valon ja hieronnan yhdistelmään, joista molempien on todettu auttavan palautumista. Ohjelmiston tarkoituksena on kyetä ohjaamaan fyysisen laitteen valoja ja tallentamaan dataa laitteen käytöstä.

1.1 Taustaa

Hierontakeinoja ja -välineitä on monenlaisia, ja yksi yleinen sekä ammattilaisten että kuluttajien kotona käyttämä keino on hierontapallo. Yksi yleisesti käytetty kotikonsti lihasjumien avaamiseksi on tennispallon pyörittely jumiutuneiden kohtien alueella. Pallo voidaan esimerkiksi asettaa maahan niin, että käyttäjä asettuu makaamaan pallon päälle jumiutuneen kohdan jääden palloa vasten. Käyttäjän liikkuessa ympäriinsä pallo liikkuu hänen allaan luoden hierovaa painetta halutulle alueelle. Mikäli pallolla halutaan hieroa kehoa muualta kuin torson alueelta, voidaan palloa painaa ja liikutella käsin hierottavaa kohtaa vasten. (Ingraham, P. 2018)

Hieronnan on todettu lisäävän nivelten liikkuvuutta, vähentävän lihasten jäykkyyttä ja parantavan verenkiertoa. Kaiken kaikkiaan hieronta kehonhuollon muotona pienentää loukkaantumisen riskiä, parantaa urheilijan suoritusta ja tehostaa palautumista. Hieronta on siis tehokas keino palautumisen tehostamiseksi, ja onkin siitä syystä lukuisten ammattiurheilijoiden suosiossa. (Weerapong, P. 2012)

1.2 Älypallon fyysinen rakenne

Kaupallisena tuotteena Älypallo koostuu fyysisestä hierontapallosta ja sitä kontrolloivasta puhelinsovelluksesta, jonka tärkeimpiä toimintoja ovat pallon valojen ohjailu ja lopullisessa versiossa diagnostiikkadatan vastaanottaminen ja esittäminen. Fyysisenä laitteena Älypallo on lihashuoltoon käytettävä hierontapallo, jossa käytetään infrapunavaloa lihasten käsittelyn tehostamiseen. Ympäri palloa on sijoiteltu NIR-valoa erittäviä lamppuja, joita ohjaillaan puhelinsovelluksella. NIR-lamppujen ja diagnostiikka-antureiden ohjailusta pallossa vastaa mikrotietokone, johon valot on kytketty ja joka yhdistää itsensä käyttäjän puhelimeen Bluetooth-yhteyden avulla. Tätä kautta sovellus lähettää pallolle komentoja ja vastaanottaa sen lähettämiä statistiikkatietoja.

NIR-valo, suomeksi lähi-infrapunasäteily, on valoa, jonka aallonpituus on välillä 700–2500 nm ja taajuus 214–400 THz. NIR-valo on infrapunavalon asteista lähimpänä näkyvää valoa, mutta infrapunavalona kuitenkin ihmissilmän havaintokyvyn ulkopuolella. Sillä on todettu olevan samankaltaisia vaikutuksia kehoon kuin hieronnalla, erityisesti verenkierron tehostamisen kautta. NIR-valo läpäisee tehokkaasti lihasta, minkä ansiosta se voi vaikuttaa helposti syvällä kehossa. (Graça, M. 2024)

Pallon toimintaperiaatteena on siis yhdistää nämä kaksi palautumista tukevaa tekniikkaa, hieronta ja NIR-valo, yhdeksi kokonaisuudeksi. Ohjainsovelluksen rooli on ohjailla hierontapallon valoja ja vastaanottaa diagnostiikkatietoja pallolta. MVP vaiheessa diagnostiikkaa ei kuitenkaan vielä kerätä, joten sovelluksen tärkein toiminto valojen ohjailun lisäksi on itsehoitojaksojen rekisteröiminen. Sovellus tarjoaa mahdollisuuden kytkeä pallon valot päälle ja pois, aloittaa tietyn kehon alueen hoitojakso, pitää kirjaa aikaisemmista hoitojaksoista ja tarkastella niiden tietoja.

2 Työn rakenne

Työ toteutetaan raporttina ohjelmistokehitysprojektista, minkä vuoksi työn rakenne poikkeaa hieman tyypillisestä tutkielmasta. Projektimuotoisessa työssä tarkastellaan pitkälti työn vaiheita ja tuloksia sen sijaan, että keskityttäisiin jonkin yksittäisen tutkimuskysymyksen tarkastelemiseen.

2.1 Vaatimusmäärittely

Työ aloitetaan vaatimusmäärittelyllä, jonka on tehnyt fyysisen laitteen toteuttaja. Vaatimusmäärittelyä arvioidaan ja laitteen toteuttajan kanssa käydään keskustelua siitä, minkä toteuttaminen on mahdollista ja mitä ei ole mielekästä lähteä vielä MVP-vaiheessa toteuttamaan. Fyysisen laitteen kehittäjän toimittama vaatimusmäärittelydokumentti keskittyy pääosin perusominaisuuksiin ja siitä käy ilmi pääpiirteissään sovelluksen halutut toiminnot ja pääasiallinen käyttökohde.

Tarkempaa määrittelyä ei ole annettu, mutta tälläkin tasolla havaitaan asioita, joiden kehittäminen ei ole vielä MVP-vaiheessa kannattavaa, kuten esimerkiksi laitteen lähettämän diagnostiikkadatan vastaanottaminen ja tulkitseminen. Lisäksi laitteen kehittäjän kanssa käydään keskustelua siitä, kannattaako vielä tässä vaiheessa sovelluksesta alkaa kehittämään iOS yhteensopivaa versiota vai ei. Kaupallista tuotetta varten tämä on olennaista, mutta esikaupallisessa vaiheessa tästä ei vielä aiheudu merkittävää etua, joten keskustelun aiheeksi otetaan, lähdetäänkö sovellusta kehittämään aluksi vain Android laitteille. (Münch, J. 2013)

Koska fyysinen laite itsessään on kehitteillä yhtä aikaa ohjainsovelluksen kanssa, on odotettavissa, että sovellukselle asetetut vaatimukset tulevat muuttumaan jonkin verran projektin aikana. Erityisesti projektin edetessä pitemmälle testaukseen on todennäköistä, että vaatimukset tulevat muuttumaan. Tähän pitää pystyä sopeutumaan, ja ohjelmiston suunnittelussa ja toteutuksessa pitää alusta alkaen ottaa huomioon mahdollisuus suurillekin muutoksille. (Lin, L. 2009)

2.2 Suunnittelu

Kun vaatimusmäärittelystä on päästy yksimielisyyteen, aloitetaan suunnitteluvaihe. Suunnitteluvaiheessa mietitään, miten vaatimusmäärittelyn mukaiset ominaisuudet saadaan parhaiten osaksi ohjelmistokokonaisuutta. Paperille luodaan malli sekä käyttöliittymästä että ohjelmiston tiedostohallinnasta. Mallien tarkoituksena on luoda pohja sille, miltä ohjainsovellus tulee näyttämään ja miten se tulee käytännössä toimimaan. Jo suunnitteluvaiheessa harjoitetaan testausta tarkastelemalla malleja siltä kannalta miten intuitiivisia ja helppokäyttöisiä mallit vaikuttavat olevan. Lisäksi suunnitelmia esitellään fyysisen laitteen kehitystiimille mielipiteitä ja palautetta varten.

Jo suunnitteluvaiheessa on tärkeä pitää mielessä, että kyseessä on hyvinvointisovellus. Kohderyhmänä toimivat hieronnan ja lihashuollon ammattilaiset, ammattiurheilijat sekä tavalliset urheilua harrastavat ihmiset. Käyttöliittymästä halutaan kohderyhmille mahdollisimman toimiva, mikä ammattilaisten tapauksessa tarkoittaa käyttömahdollisuuksien laajentamista ja palautteena saadun datan selkeyttä sekä monipuolisuutta. Tavallisten käyttäjien tarpeet ovat yksinkertaisemmat, ja tämän vuoksi käyttökokemuksesta halutaan mahdollisimman intuitiivinen ja helppokäyttöinen. (Cotton, V. 2019) (Yin, S. 2022)

Myös vaatimusmäärittelyn muutokset projektin edetessä ja projektin jatkaminen kohti kaupallista vaihetta MVP-tasoisen tuotteen jälkeen on otettava huomioon. Ohjelmistoa tulee kyetä tulkitsemaan ja muokkaamaan sekä jatkokehittämään helposti, jotta siitä olisi projektille todellista hyötyä. Jos ohjelmisto jää MVP-tasoiseksi raakileeksi ilman todellista jatkokehittämisen mahdollisuutta joudutaan kaupalliseen vaiheeseen siirryttäessä ohjelmiston kehittäminen aloittamaan alusta, mikä olisi silkkaa resurssien tuhlausta. (Kazakevich, B. 2024)

2.3 Kehitys

Suunnittelusta siirrytään toteutukseen, jossa aloitetaan ohjelmiston kehitys tehtyjen suunnitelmien pohjalta. Kehitys tapahtuu pääosin Android Studio -kehityssovelluksella, joka keskittyy nimenomaan Android laitteille kehitettäviin sovelluksiin. Tämän lisäksi yksittäisiä osuuksia on mahdollista kehittää muilla kehitysalustoilla, kuten Visual Studio Codella. Versionhallintaan käytetään GitHubia.

Kehitystyössä pääkohtia ovat käyttöliittymä, sovelluksen sisäinen logiikka sekä kommunikaatio Älypallon kanssa. Kommunikaatio tapahtuu Bluetooth-yhteyden avulla välitetyillä komennoilla sovelluksen lähettäessä käskyjä Älypallolle ja Älypallon lähettäessä diagnostiikkadataa sovellukselle. MVP-vaiheessa diagnostiikkadataa ei vielä kerätä, joten kommunikaatio tulee koostumaan yksinomaan Älypallolle lähetettävistä komennoista, mutta jatkokehitystä ajatellen datan vastaanottaminen pyritään tekemään mahdollisimman helpoksi jo tässä vaiheessa.

Kehitystyö jakautuu viikon mittaisiin sprintteihin, joiden aikana kehitetään yksittäistä ominaisuutta tai toimintoa. Sprintin päätteeksi työn tuotosta arvioidaan sekä yksikseen että osana ohjelmistokokonaisuutta. Lisäksi työn edistymistä esitellään fyysisen laitteen kehittäjätiimille. Siinä vaiheessa, kun fyysinen laite pääsee tilanteeseen, jossa se kykenee vastaanottamaan komentoja sovellukselta, yhteistyö laitteen kehitystiimin kanssa syvenee ja sprinttien merkitys vähenee kehitystyön virtaviivaistuessa (Evans, I. 2004). (Harned, D. 2018)

Mikäli fyysisen laitteen kehityksessä ilmenee ongelmia, pyritään sovelluksen kehitystä jatkamaan mahdollisuuksien mukaan tästä huolimatta. Bluetooth-yhteyden korvaaminen mockeilla saattaa tulla ajankohtaiseksi, jos laitteeseen yhdistäminen Bluetoothilla ei ole mahdollista tämän projektin aikana. Tällöin sovellus pyritään kehittämään mahdollisimman valmiiksi ilman yhteyttä fyysiseen laitteeseen, mikä tarkoittaa ohjelmiston sisäisen toiminnan nousemista kehitystyön korkeimmaksi prioriteetiksi. (Mortensen, M. 2008)

2.4 Testaus

Testausta toteutetaan hyvän ohjelmistokehitysperiaatteen mukaisesti halki koko kehitysprojektin. Konseptien ja suunnitelmien testaaminen aloitetaan jo suunnitteluvaiheessa, ja testausta jatketaan koko projektin ajan sitä mukaa kun uusia ominaisuuksia saadaan testikuntoisiksi. MVP-tasoisen ohjelmiston testauksessa otetaan huomioon tuotteen keskeneräinen luonne, ja tässä vaiheessa testaus keskittyykin pääasiallisesti Älypallon kehitystiimin omaan testaukseen. Erityisesti käyttäjäkokemusta testataan lisää jatkokehityksen yhteydessä tuotteen lähestyessä kaupallista vaihetta. (Leloudas, P. 2023)

Ohjelmiston kehittäjä toteuttaa jatkuvaa testausta kehityksen yhteydessä yksikkötestien muodossa, minkä lisäksi kehitystiimin muita jäseniä otetaan testaamaan ohjelmistoa

kokonaisuutena mustan laatikon periaatteella. Näin saadaan varmistettua toiminnallisuus esikaupallisen vaiheen tärkeimpänä ominaisuutena, sillä käyttäjäkokemus ja sen myötä käyttöliittymän merkitys korostuvat vasta kaupallisen tuotteen kanssa. MVP-tuote on kokonaisuus, eikä sitä saa jättää pelkän toiminnallisuuden tasolle, mutta testauksessa se tulee olemaan keskiössä. (Cheng, L. 2016)

Kun koetaan, että sovelluksen kehityksessä on saavutettu tarpeeksi suuri toiminnallisuus, siirtyy keskittyminen kehityksessä ja näin ollen myös testauksessa käyttöliittymään ja sitä myöten helppokäyttöisyyteen ja intuitiivisuuteen. Intuitiivisuuden kanssa on tärkeä ymmärtää, että kehittäjä, joka on jatkuvasti tekemisissä tuotteensa kanssa, sokeutuu tuotteen käytettävyydelle, ja näin ollen on olennaista testata käyttöliittymän helppokäyttöisyyttä ulkopuolisilla henkilöillä, jotka eivät ole työskennelleet ohjelmiston parissa. (Paladino, N)

3 Projektin toteutus

Tässä luvussa tarkastellaan fyysisen laitteen kehittäjältä saatua vaatimusmäärittelyä, siitä käytyä keskustelua sekä ohjelmiston toteuttamiseksi tehtyjä suunnitelmia. Vaatimusmäärittelyn osalta käsittely on vähäistä, sillä projekti kehittyy jatkuvasti työn edetessä, eikä laajaalaista vaatimusmäärittelyä ole näin ollen helppoa tai edes järkevää kasata.

3.1 Vaatimusmäärittelyn muutokset

Vaatimusmäärittelyn alkuperäisiin kohtiin lukeutui muun muassa iPhone yhteensopivuus sekä keskusteluyhteys fysioterapeutin kanssa. Nämä ja muita kohtia kuitenkin jätettiin MVP-tason toteutuksesta pois käytännöllisistä syistä, sillä kumpikaan ei ole olennainen pienimmän toimivan tuotteen periaatteiden kannalta. Ominaisuudet halutaan kyllä lopulliseen tuotteeseen, sillä ne koetaan kaupallisen vaiheen kannalta korvaamattomiksi. Erityisesti mahdollista käyttäjäkuntaa lisäävä iPhonelle sopiva versio koetaan tärkeäksi tulevaisuuden kannalta.

3.1.1 Vaatimusmäärittelyn sisältö

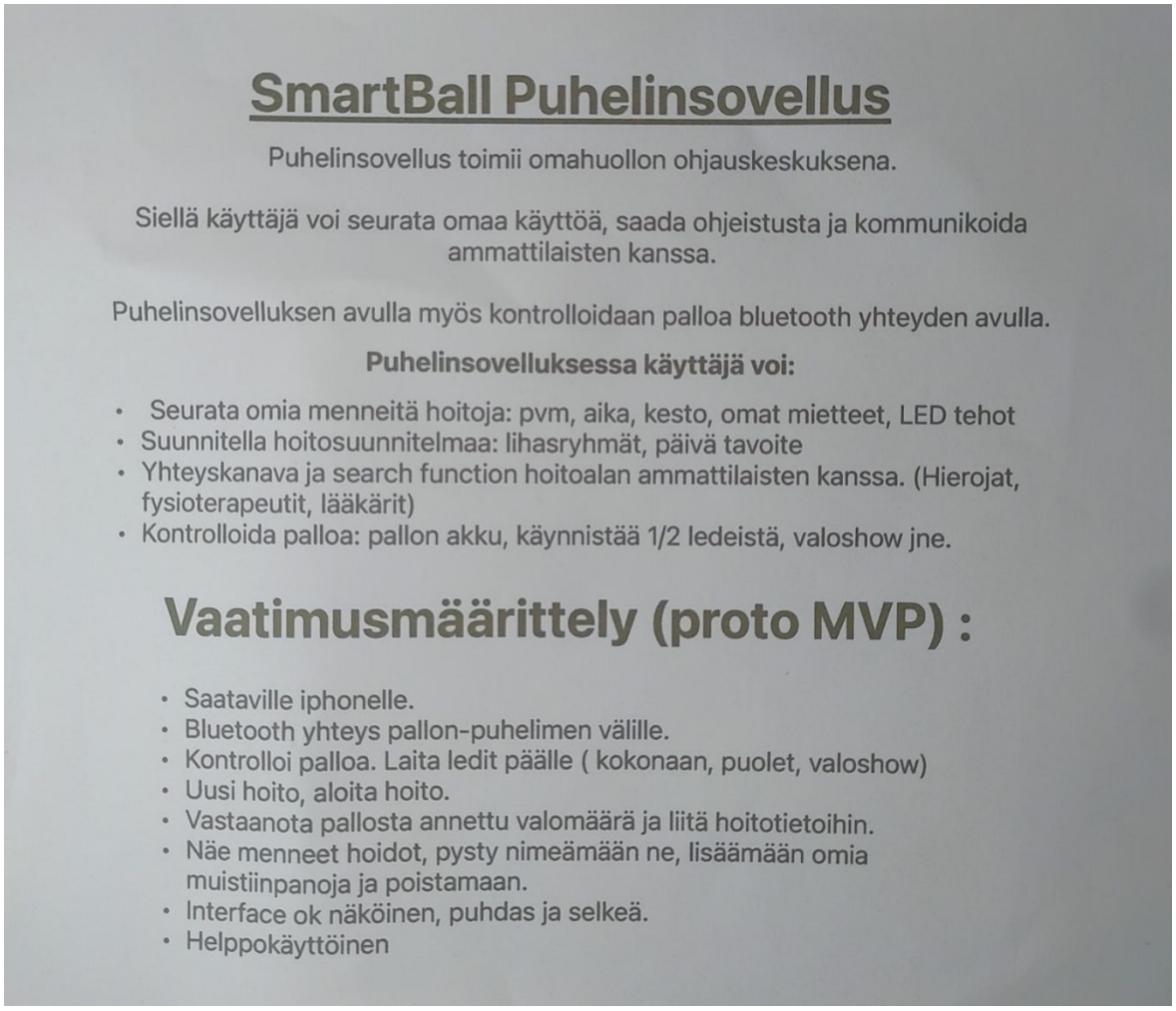
Alkuperäinen vaatimusmäärittelydokumentti (kuva 1) sisältää fyysisen laitteen toiminnan kannalta olennaisia ominaisuuksia, kuten palloon yhdistäminen, valojen ohjailu ja hoitohistorian tarkastelu. Dokumentissa sovellusta kuvaillaan seuraavasti: ”Puhelinsovellus toimii omahuollon ohjauskeskuksena. Siellä käyttäjä voi seurata omaa käyttöä, saada ohjeistusta ja kommunikoida ammattilaisten kanssa.”

Varsinaisia vaatimuksia ja toiminnallisuuksia on lueteltuna seuraavasti:

- Saatavilla iPhonelle
- Bluetooth yhteys pallon-puhelimen välille
- Kontrolloi palloa. Laita ledit päälle (kokonaan, puolet, valoshow)
- Uusi hoito, aloita hoito

- Vastaanota pallosta annettu valomäärä ja liitä hoitotietoihin
- Näe menneet hoidot, pysty nimeämään ne, lisäämään omia muistiinpanoja ja poistamaan
- Interface ok näköinen, puhdas ja selkeä
- Helppokäyttöinen

Lisäksi ennen vaatimuksien listausta listataan toimintoja, jotka sovellukseen halutaan. Toimintoihin lukeutuvat menneiden hoitojen seuraaminen, hoitosuunnitelman tekeminen, yhteyskanava hoitoalan ammattilaisiin ja fyysisen laitteen kontrolloiminen. Näitä on kutakin avattu esimerkeillä toiminnallisuuden selventämiseksi.



SmartBall Puhelinsovellus

Puhelinsovellus toimii omahuollon ohjauskeskuksena.

Siellä käyttäjä voi seurata omaa käyttöä, saada ohjeistusta ja kommunikoida ammattilaisten kanssa.

Puhelinsovelluksen avulla myös kontrolloidaan palloa bluetooth yhteyden avulla.

Puhelinsovelluksessa käyttäjä voi:

- Seurata omia menneitä hoitoja: pvm, aika, kesto, omat mietteet, LED tehot
- Suunnitella hoitosuunnitelmaa: lihasryhmät, päivä tavoite
- Yhteyskanava ja search function hoitoalan ammattilaisten kanssa. (Hierojat, fysioterapeutit, lääkärit)
- Kontrolloida palloa: pallon akku, käynnistää 1/2 ledeistä, valoshow jne.

Vaatimusmäärittely (proto MVP) :

- Saataville iphonelle.
- Bluetooth yhteys pallon-puhelimen välille.
- Kontrolloi palloa. Laita ledit päälle (kokonaan, puolet, valoshow)
- Uusi hoito, aloita hoito.
- Vastaanota pallosta annettu valomäärä ja liitä hoitotietoihin.
- Näe menneet hoidot, pysty nimeämään ne, lisäämään omia muistiinpanoja ja poistamaan.
- Interface ok näköinen, puhdas ja selkeä.
- Helppokäyttöinen

Kuva 1. Vaatimusmäärittelydokumentti

Dokumentti on itsessään vaatimaton, mutta siitä käy kuitenkin ilmi sovelluksen päätavoitteet ja yleinen suunta, johon sovellusta halutaan lähteä rakentamaan.

3.1.2 Keskustelu vaatimusmäärittelystä

Vaatimusmäärittelyssä havaituista ongelmista keskusteltiin vaatimusmäärittelyn tekijän kanssa. Ensimmäinen keskusteltu asia oli saatavuus iPhoneille, jonka laitteen kehittäjä haluaa laajentaakseen tuotteen kohderyhmää ja tuodakseen sen saataville mahdollisimman monelle. Saatavuus iPhoneille on kuitenkin MVP-vaiheen kehityksen ja testauksen yhteydessä tarpeetonta, sillä laitteesta voidaan rakentaa täysin toimiva versio, joka toimii yhdessä Android älypuhelimelle asennetun ohjainsovelluksen kanssa, ja toteuttaa näin MVP-tuotteen. Luonnollisesti iPhone yhteensopivuus pitää kehittää ja testata erikseen kaupallista vaihetta varten, mutta tässä kohtaa sovelluksen kehittäjän vähäinen kokemus iPhoneille kehitettävistä sovelluksista tekee Android sovelluksen kehittämisestä huomattavasti helpompaa ja nopeampaa. Kun asia oli selitetty vaatimusmäärittelyn esittäjälle, tämä ymmärsi tilanteen ja hyväksyi kehityksen ensin pelkästään Android älypuhelimille, joten iPhone kehitys jätettiin pois suunnitelmista. Hän kuitenkin painotti iPhone yhteensopivuuden tarvetta ja merkitystä kaupallisen vaiheen kannalta, ja iPhonelle sopivan version kehittäminen tuleekin olemaan tärkeä prioriteetti jatkokehityksessä.

Toinen keskusteltu aihe oli kommunikaatioyhteys hoitoalan ammattilaisiin. Tämä ominaisuus ei ole sovelluksen toiminnan ytimessä, ja näin ollen MVP-tasoinen tuote voidaan rakentaa ilman kommunikaatiotoimintoa. Erityisesti tämä pitää paikkansa siksi, että esikaupallisessa vaiheessa hoitoalan ammattilaisen saaminen kommunikaatiokanavan toiseen päähän nähtiin tarpeettoman vaivalliseksi. Vaatimusmäärittelyn toimittaja oli näistä sekoista samaa mieltä, ja ominaisuus oli myös hänen mielestään pikemminkin pitkän tähtäimen tavoite kuin MVP-tason olennainen toiminto. Näin ollen myös sen kehittäminen jätettiin toistaiseksi pois suunnitelmista, mutta siihenkin tullaan palaamaan sovelluksen jatkokehityksen yhteydessä.

3.2 Suunnitelmat

Sovelluksen suunnittelu aloitettiin hahmottelemalla käyttöliittymää paperille. Näitä paperisia suunnitelmia tarkasteltiin ja jatkokehitettiin havaittujen haasteiden pohjalta. Sovelluksen

toiminnallisuutta pyrittiin suunnittelemaan esimerkiksi käyttötapausten, käyttötapauskaavioiden ja sekvenssikaavioiden avulla.

3.2.1 Paperiset käyttöliittymämallit

Paperille hahmoteltujen käyttöliittymien pyrkimyksenä oli olla ensimmäinen hahmotelma intuitiiviselle ja interaktiiviselle käyttöliittymälle. Ensimmäisessä suunnitelmassa hahmoteltiin perinteistä fragment -näkymää, jossa käyttäjä voi navigointipalkin turvin vaihtaa näkymän oleellisten toimintojen mukaiseksi. Toiminnoiksi ajateltiin vaatimusmäärittelyn mukaisesti hoitonäkymää, joka oli myös hahmotelmassa avoinna oleva näkymä ja josta käyttäjä voi aloittaa hoidon, sekä hoitohistoria, jonka tarkoituksena on esitellä käyttäjälle tämän aikaisempia hoitoja.

Seuraavassa mallissa (kuva 2) oli lisätty kolmas näkymä asetuksia varten sekä toimintoja avoinna olevalle hoitonäkymälle.

Kuva 2. Käyttöliittymäsuunnitelma

Toimintojen tarkoituksena on antaa käyttäjän kontrolloida aloitettavaa hoitoa, ja säädettäviksi asetuksiksi on asetettu hoidon kesto ja hierottava kehonosa. Lisäksi hoitohistorialle oli

kehitetty suunnitelma, josta ilmenee hoitojen näkyminen vieritettävässä valikossa, sekä hoidoista näytettävät tiedot: hoidon kesto, hierottu alue ja hoidon kellonaika.

3.2.2 Toimintokaaviot

Ohjelmiston toiminnallisuuden mallintamiseen käytettiin UML-kaavioita, jotka kuvasivat ennen kaikkea yksittäisiä käyttötapauksia. Näitä esimerkinomaisia kaavioita käytettiin ohjelmiston perustoimintojen hahmottamiseen. Seuraavanlaisia käyttötapauksia käytettiin hahmottamaan ohjelmiston toimintaa:

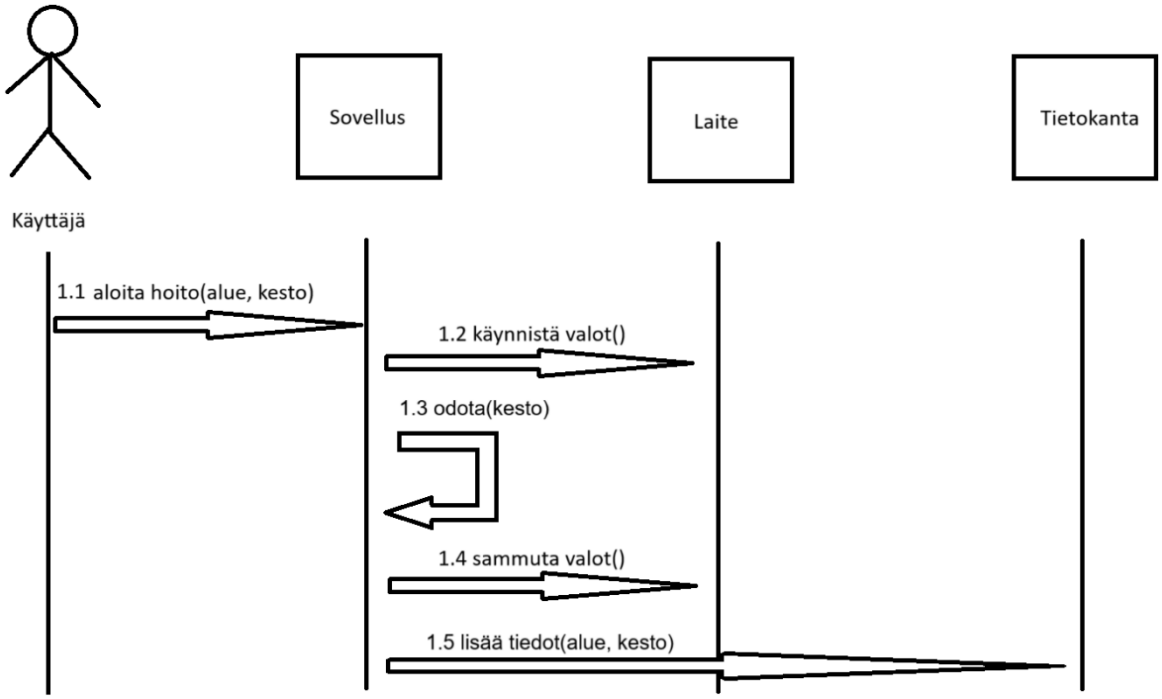
Taulukko 1. Käyttötapaukset

Nro.	Lähtötilanne	Käyttötapaus
1	Hoitosivu auki, pallo käynnissä ja yhdistettynä puhelimeen	Käyttäjä painaa aluevalikosta valitakseen hierottavan kehon alueen. Esiin ilmestyy lista kehon osista, ja käyttäjä valitsee pohkeen. Käyttäjä painaa aikavalikosta valitakseen hoidolle keston, ja esiin ilmestyy lista mahdollisista kestoista, joista käyttäjä valitsee kymmenen minuuttia. Lopuksi käyttäjä painaa hoidon aloittavaa nappia, jonka jälkeen sovellus käskee palloa aktivoitumaan, näytölle ilmestyy ajastin sekä hierontaohjeita, ja käyttäjä siirtyy hieromaan pohjettaan.
2	Hoito käynnissä ja pallo käynnissä ja yhdistettynä puhelimeen	Käyttäjä painaa hoidon aikana hoidon keskeyttävää nappia, jolloin käynnissä olevan hoidon ikkuna sulkeutuu. Jos hoito kesti yli 30 sekuntia, sen tiedot kirjataan hoitohistoriaan. Sovellus avaa hoitoikkunan, jossa näkyvät tämän edelliset asetukset, josta käsin asetuksia voi säätää ja hoidon aloittaa uudestaan.
3	Hoitosivu auki	Käyttäjä painaa hoitohistorian avaavaa nappia, ja näkymä siirtyy hoitosivulta hoitohistoriaan. Kaikki kirjatut hoidot esitetään selattavassa listassa, josta hoidon tietoja voi tarkastella ja hoitoja muokata. Käyttäjä valitsee viimeisimmän hoidon, ja sen sivu aukeaa näytölle esitellen tiedot ja muokattavat kohdat. Käyttäjä painaa hoidon poistavaa nappia. Sovellus varmistaa käyttäjältä,

		haluaako tämä varmasti poistaa hoidon. Käyttäjä painaa kyllä, ja valitun hoidon tiedot poistuvat tietokannasta ja hoito itse listasta.

Näiden käyttötapausten tarkoituksena on hahmottaa sovelluksen tavallista käyttöä. Käyttötapauksissa käsitellään uuden hoidon asetusten säätäminen ja aloittaminen, hoidon keskeyttäminen sekä hoitohistorian tarkasteleminen ja hoitojen poistaminen. Tällainen käyttötapausten kokonaisuus voisi tapahtua esimerkiksi silloin, kun käyttäjä aloittaa hoidon, mutta joutuukin lopettamaan sen pian aloittamisen jälkeen, eikä halua lyhyttä, käytännössä merkityksetöntä hoitoa historiaansa täyttämään.

Sekvenssikaavioilla (kuva 3) mallinnettiin sovelluksen sisäistä logiikkaa. Pyrkimyksenä oli esittää sovelluksen perustoiminnallisuus, hoidon aloittaminen ja kirjaus, järkevässä muodossa, tapahtuma kerrallaan kronologisessa järjestyksessä.



Kuva 3. Sekvenssikaavio

Hoidon aloittamisen tapauksessa käyttäjä antaa sovellukselle käskyn aloittaa hoito, ja tämän käskyn mukana välitetään tiedot hierottavasta alueesta ja hoidon kestosta. Sovellus välittää hierontapallolle käskyn käynnistää valot, odottaa käyttäjän antaman ajan ja lähettää sitten pallolle käskyn sammuttaa valot. Viimeiseksi sovellus lähettää tietokannalle tiedot hoidosta.

Tämä on hoidon aloittamisen perusmalli, jonka pohjalta lähdettiin seuraavassa vaiheessa kehittämään ohjelmiston toiminnallisuutta.

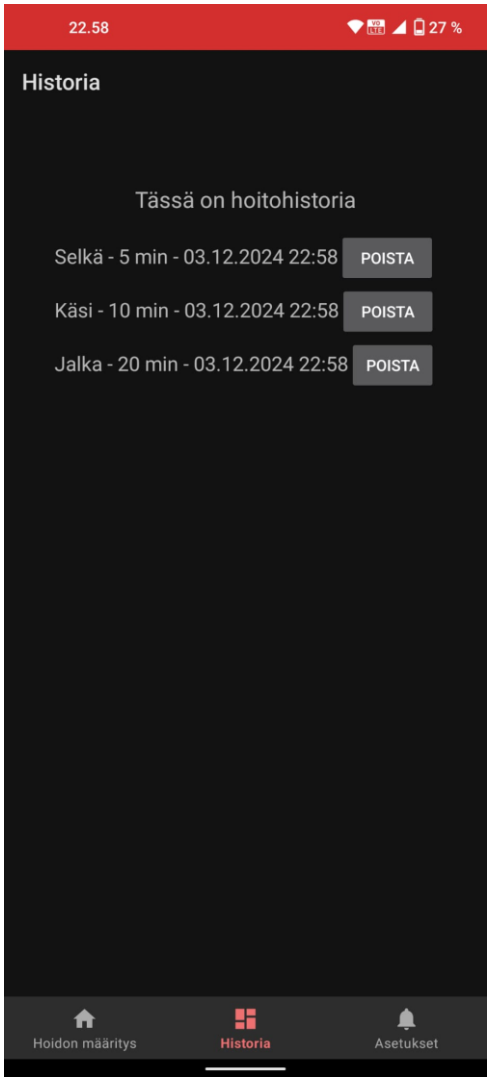
3.3 Kehitystyö

Kehitystyö tapahtui käytännössä kokonaan Android Studiolla, Visual Studio Coden käyttö oli loppujen lopuksi minimaalista. Työ alkoi käyttöliittymäelementtien sijoittelulla Android Studion Design -toiminnolla, joka antaa sijoitella elementtejä raahaamalla niitä ympäriinsä, ja siitä siirryttiin toiminnallisuuden ohjelmointiin. Ohjelmointikielenä on käytetty Javaa SQL:llä rakennettua tietokantaa lukuun ottamatta. Kehitystyön edistyessä jokainen uusi toiminnallisuus, korjaus ja muutos päivitettiin säännöllisesti GitHubiin, jota käytettiin versionhallintaan.

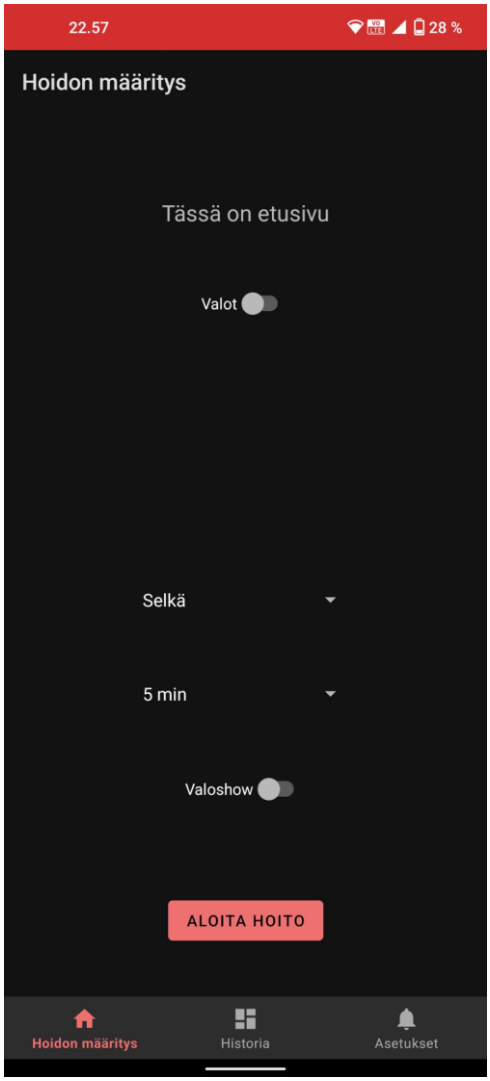
3.3.1 Käyttöliittymä

Käyttöliittymän rakentaminen aloitettiin suunnitelmien mukaisesti hoitosivun rakentamisesta. Pohjana käytettiin fragment -pohjaa, joka luo heti projektin alkuun näytön alalaitaan navigointipalkin, jota käyttämällä pääsee vaihtamaan sivujen välillä. Navigointipalkin yläpuolelle asetettiin käyttöliittymäsuunnitelman mukaisesti nappi hoidon aloittamiselle. Napin yläpuolelle sijoitettiin kaksi kappaletta vetovalikkoja, toinen hoidon kestolle ja toinen hierottavalle alueelle. Nämä kaikki ankkuroitiin keskelle näyttöä, ja kunkin elementin alalaita ankkuroitiin sopivan etäisyyden päähän alemman elementin ylälaidasta. Näin saatiin aikaan kiinteä kokonaisuus, joka asettuu luontevasti näytön alalaidasta keskipaikkeille.

Lisäksi sopivalle etäisyydelle näytön ylälaidasta asennettiin vielä kytkin pallon valojen kontrolloimiselle, joka mahdollistaisi valojen itsenäisen toiminnan suunnitellun hoidon ulkopuolella. Lisäksi hoidon käynnistävän napin ja vetovälikköjen väliin sijoitettiin valoshow kytkin, joka oli mainittu vaatimusmäärittelyssä samassa kohtaa valot itsenäisesti käynnistävän toiminnon kanssa. Valoshow kytkimelle ei ole ohjelmoitu tarkempaa toiminnallisuutta, ja onkin mahdollista, että se tulee vaihtamaan nimeä tai katoamaan kokonaan jatkokehityksen yhteydessä.



Kuva 4. Hoitohistoria



Kuva 5. Hoitosivu

Hoitohistorian käyttöliittymä seurasi myös pitkälti suunnitelmia, ja näin ollen toistaiseksi ainoa sisältö sivulla on rullattava näkymä, jonne listataan tiedot menneistä hoidoista. Hoidoista esitellään kaikki olennaiset tallennetut tiedot, eli hierottu alue, hoidon kesto sekä hoidon ajankohta. Lisäksi jokaisen kirjauksen yhteyteen lisättiin nappi, jota painamalla käyttäjä voi varmistusviestin jälkeen poistaa kyseisen kirjauksen hoitohistoriasta.

Asetussivu on olemassa, mutta siellä ei vielä ole sisältöä. Käyttöliittymän värimaailmaksi valikoitui laitteen kehittäjän kanssa käytyjen keskustelujen myötä trendikäs mustan ja punaisen yhdistelmä. Jatkokehitystä ajatellen valoisa vaihtoehto väriteemaksi, mikä onkin nykyään saatavilla useimmissa sovelluksissa, lienee hyvä ajatus.

3.3.2 Toiminnallisuus

Toiminnallisuuden rakentaminen aloitettiin hoitohistorian kirjaston rakentamisesta, sillä tässä kohtaa kehitystä oli ilmennyt, että fyysiseen laitteen kehityksen kanssa oli syntynyt ongelmia, ja näin ollen yhteyden muodostaminen laitteeseen tulisi järjestymään vasta myöhemmässä vaiheessa. Toisin sanoen ohjainsovelluksen ja laitteen välistä toiminnallisuutta ei voitu vielä kehittää, joten keskityttiin sovelluksen sisäiseen toimintoon, hoitohistoriaan.

Hoitohistorian tarkoituksena on säilyttää käyttäjän tekemät hoidot ja esittää ne käyttäjälle järkevällä tavalla. Tämän toteuttamiseksi yksittäiselle hoidolle luotiin luokka, johon hoidon yksityiskohtaiset tiedot voidaan tallentaa, ja hoitojen säilyttämistä varten SQL tietokanta. Hoidon aloittavaan nappiin tehtiin toiminnallisuus, joka ottaa tiedot ajan ja hierottavan alueen vetovalikoiden valinnoista, sekä senhetkisen kellonajan, ja luo näistä tiedoista hoito-olion, joka tallennetaan tietokantaan. Hoitohistoria puolestaan hakee koko tietokannan sisällön, tutkii sen jokaisen olion ja luo olioiden tietojen pohjalta edellä kuvatun mukaisen listan hoidoista.

Koska fyysiseen laitteeseen ei ole vielä ollut mahdollista yhdistää, on sovellukseen luotu yhteyttä valmistelevaa toiminnallisuutta. Hoidon aloittava nappi ja valot päälle kytkevä kytkin kykenevät lähettämään käskyjä Bluetooth yhteyden kautta, mutta käskyjen tarkasta sisällöstä ei ole vielä varmuutta eikä niiden toimivuutta ole siksi voitu myöskään testata.

3.4 Testaaminen

Testaamista harjoitettiin säännöllisesti läpi projektin. Pääosa testaamisesta koostui uusien toiminnallisuuksien testaamisesta yksikkötestien muodossa, mutta suurimmat yksittäiset testikokonaisuudet muodostuivat sovelluksen esittelemisestä fyysisen laitteen kehitystiimille. Yksikkötestien tarkoituksena oli ennen muuta varmistaa, että uudet elementit toimivat halutulla tavalla, kun taas esittelykierrosten tarkoituksena oli testauttaa ohjelmistoa kokonaisuudessaan ihmisillä, jotka eivät olleet sen kanssa tekemisissä päivittäin.

Tärkein anti testauksessa oli yksittäisten toimimattomien osien havaitsemisen lisäksi kehityksen suunnan tarkastelu. Esimerkiksi sovelluksen värimaailma ja hoidon poistamisen

yhteydessä ilmestyvä varmistusikkuna ovat ominaisuuksia, jotka lisättiin ohjelmistoon kehitystiimin antaman palautteen myötä.

Näin pienen, MVP-tasoisen projektin aikana toiminnallisuutta ei ole ehtinyt syntyä vielä niin paljoa, että ison skaalan testausta pystyisi järkevästi toteuttamaan. Ohjelmiston toiminnallisuus monimutkaistuu huomattavasti, kun yhteys fyysiseen laitteeseen on tulevaisuudessa saavutettu, ja siinä kohtaa myös testauksen taso ja merkitys tulevat kasvamaan huomattavasti. Siinä vaiheessa myös yhteistyö muun kehitystiimin kanssa tulee lisääntymään, mikä johtanee siihen, että sovellus tulee kehitystiimin jäsenille tutuksi, ja näin ollen on hyvä etsiä testihenkilöitä, joille sovellus ei ole tuttu. Tämä helpottaa sovelluksen käyttöliittymän ja toiminnallisuuksien intuitiivisuuden tarkastelua.

4 Lopputulema

Fyysisen laitteen kehityksen yhteydessä ilmenneitten ongelmien takia ohjainsovellus ei valitettavasti ole vielä saavuttanut tavoitettaan. Sovellus on kuitenkin kehitetty mahdollisimman pitkälle ilman yhteyttä fyysiseen laitteeseen, ja yhdistämistä on valmisteltu, jotta toiminta fyysisen laitteen kanssa olisi mahdollisimman helppoa, kun se tulevaisuudessa mahdollistuu. Sovellus on siis valmisteltu toimimaan tarkoituksensa mukaisesti niin pitkälle kuin projektin kokonaistilanne sen sallii.

Fyysiseen laitteeseen yhdistäminen tulee olemaan tulevan kehityksen ensimmäinen ja tärkein kohta, sillä sen myötä sovellus tulee viimein saavuttamaan haetun MVP-tason. Tämän jälkeen kehityskohteina tullaan pitämään olemassa olevien toimintojen hiomista sekä fyysisen laitteen lähettämän diagnostiikkadatan vastaanottamista, tallettamista ja esittämistä osana hoitohistoriaa. Myöhemmässä vaiheessa kommunikaatio hoitoalan ammattilaisten kanssa tullaan myös lisäämään. Lisäksi suuri kehityskohde tulee olemaan iPhone yhteensopiva versio sovelluksesta, mutta se ei enää ole tämän kyseisen sovelluksen jatkokehittämistä vaan pikemminkin kokonaan erillinen projektinsa. Ohjelmistokehitysprojektien yhteydessä on tavallista, että alkuperäinen budjetti ja aikataulu ovat riittämättömiä, ja tämä johtaa usein ongelmiin projektin edetessä (Deschamps, T. 2024).

Tämän projektin kehitystyö on kokonaisuudessaan onnistunut odotusten ja suunnitelmien mukaisesti. Erinäisiä haasteita kuitenkin nousi esiin, suurimpana näistä jo mainitut haasteet fyysisen laitteen kehityksen kanssa ja niistä seuranneet ongelmat ohjelmiston kehittämisessä. Tämä on haaste, joka syntyi ulkoisen uhan seurauksena, mutta jonka vaikutuksia projektille pystyttiin onneksi minimoimaan asiaan varautumisen ansiosta. (Lin, L. 2009)

Lähteet

Cheng, L. C. (2016) ‘The Mobile App Usability Inspection (MAUi) Framework as a Guide for Minimal Viable Product (MVP) Testing in Lean Development Cycle’, in *Proceedings of the 2nd International Conference in HCI and UX Indonesia 2016*. [Online]. 2016 New York, NY, USA: ACM. pp. 1–11.

Cotton, V. and Patel, M.S. (2019) ‘Gamification Use and Design in Popular Health and Fitness Mobile Applications’, American journal of health promotion: AJHP, 33(3), ss. 448–451. Saatavissa: https://doi.org/10.1177/0890117118790394.

Deschamps, T. (2024) Workers warn of tight project deadlines: Software experts say pressure to deliver on time, on budget puts safety at risk.

Evans, I. (2004) *Achieving software quality through teamwork*. 1st ed. Boston: Artech House.

Graça, M. F. P. et al. (2024) Application of near-infrared light responsive biomaterials for improving the wound healing process: A review. *Journal of drug delivery science and technology*. [Online] 93105409-.

Harned, D. (2018) *Hands-on agile software development with JIRA: design and manage software projects using Agile methodology*. Birmingham: Packt.

Ingraham, P. (2018). *Tennis Ball Massage for Myofascial Trigger Points*. [online] painscience.com. Available at: https://www.painscience.com/articles/tennis-ball.php [Accessed 30 Oct. 2024].

Kazakevich, B. & Joiner, K. (2024) Agile approach to accelerate product development using an MVP framework. *Australian journal of multi-disciplinary engineering*. [Online] 20 (1), 1–12.

Leloudas, P. (2023) ‘The Importance of Software Testing’, in *Introduction to Software Testing*. [Online]. United States: Apress L. P. pp. 1–4.

Lin, L. et al. (2009) The impact of requirements changes on specifications and state machines. *Software, practice & experience*. [Online] 39 (6), 573–610.

Mortensen, M. et al. (2008) A test driven approach for aspectualizing legacy software using mock systems. *Information and software technology*. [Online] 50 (7), 621–640.

Münch, J. et al. (2013). Creating Minimum Viable Products in Industry-Academia https://doi.org/10.1007/978-3-642-44930-7_9

Paladino, N. (n.d.). *3 Strategies to Overcome Developer Bias and Product Bias*. [online] Blue Triangle. Available at: https://bluetriangle.com/blog/3-strategies-to-overcome-developer-bias-and-product-bias [Accessed 25 Nov. 2024].

Weerapong, P. et al. (2005) The mechanisms of massage and effects on performance, muscle recovery and injury prevention. *Sports medicine (Auckland)*. [Online] 35 (3), 235–256.

Yin, S. et al. (2022) Impact of gamification elements on user satisfaction in health and fitness applications: A comprehensive approach based on the Kano model. *Computers in human behavior*. [Online] 128107106-.