

Toni Kukkohovi

**TUTKIMUSPÄIVÄKIRJA-MOBIILISOVELLUKSEN OMINAISUUK-
SIEN JA VAATIMUSTEN SELVITYS SEKÄ SOVELLUSKEHITYS**

TUTKIMUSPÄIVÄKIRJA-MOBIILISOVELLUKSEN OMINAISUUK- SIEN JA VAATIMUSTEN SELVITYS SEKÄ SOVELLUSKEHITYS

Toni Kukkohovi
Opinnäytetyö
Syksy 2020
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä: Toni Kukkohovi

Opinnäytetyön nimi suomeksi: Tutkimuspäiväkirja-mobiilisovelluksen ominaisuuksien ja vaatimusten selvitys sekä sovelluskehitys

Opinnäytetyön nimi englanniksi: Research diary -mobile application's features and requirements clearance and application development

Työn ohjaajat: Anne Keskitalo, Mika Mustonen

Työn valmistumislukukausi ja -vuosi: Lokakuu 2020

Sivumäärä: 39

Tämän opinnäytetyön tarkoituksena oli tehdä järjestelmäkokonaisuuden ja arkkitehtuurin selvitys tutkimuspäiväkirja mobiilisovelluksen käytöstä. Polarin tutkustiimi on aikaisemmin järjestänyt erilaisia yrityksen sisäisiä ja ulkopuolisia tutkimuksia, kuten uni- ja päiväkirjatutkimuksia. Näissä tutkimuksissa on kuitenkin ollut ongelmana tutkittavien tekemien kirjausten huono kattavuus sekä vastauksien haasteellinen raportointi.

Opinnäytetyön aikana perehdyttiin tutkimuksien käyttötarkoitukseen, järjestelmäkokonaisuuden hahmottamiseen ja mobiilisovelluksen kehittämiseen.

Työn lopputuloksena syntyi kattava raportti, joka kuvaa, mitä varten tällaista järjestelmää tarvittaisiin, mistä eri komponenteista järjestelmä koostuu ja mitä ominaisuuksia käyttöönottoa ja järjestelmäkokonaisuutta varten tarvitsee kehittää. Raportin lisäksi työssä kehitettiin alustava Android-pohjainen mobiilisovellus, jonka avulla on helppo hakea eri näkökulmia järjestelmän suunnittelulle ja toteutukselle. Sovelluksen ansiosta tulevaisuudessa on helpompi lähteä pilotoimaan ja jatkokehittämään järjestelmää.

Asiasanat: urheiluteknologia, mobiilisovellukset, Android-ohjelmointi, tutkimuspäiväkirja, ohjelmistokehitys, urheilukellot

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Software Development

Author: Toni Kukkohovi

Title of thesis: Research diary -mobile application's features and requirements clearance and application development

Supervisor: Anne Keskitalo, Mika Mustonen

Term and year when the thesis was submitted: October 2020

Pages: 39

The goal of this thesis was to design a research diary system and architecture but also to implement an Android-based mobile application for the system. Polar's research team has organized different kinds of researches where they kept analysis with sleep- and diary researches. These previous researches had problems with the quality and coverage of the self-reported data and subjects reporting the results.

During the thesis, I resolved and figured out the intended use of the research application and system. Work included interviews and developing an Android mobile application demo of the system.

The result of the thesis was a comprehensive report describing what research diary system would be needed for, what different kinds of components the system consists of, and what features should be developed for the implementation and the system. In addition to the report, an Android-based mobile application demo was developed. With the demo application, it is easy to search for different perspectives for the user interface design and feature implementation. The demo application will also make it easier to develop and test the system further in the future.

Keywords: sports technology, mobile development, Android-programming, research diary, software development, sportwatch

ALKULAUSE

Tämän opinnäytetyön tilaajana toimi Polar Electro Oy. Polarilta työn ohjaajana toimi tuote- ja palvelupuolen ohjelmistokehitys tiimien johtaja Mika Mustonen. Oulun ammattikorkeakoululta opinnäytetyön ohjaajana toimi tietotekniikan lehtori Anne Keskitalo. Haluaisin kiittää kaikkia työssä auttaneita ja mukana olleita henkilöitä sekä erityisesti haastattelemani henkilöitä hyvistä ideoista ja neuvoista. Iso kiitos myös erityisasiantuntija Jukka Haposelle palvelinpuolen konfiguroinnista ja työhön liittyvistä neuvoista.

Oulussa 3.11.2020

Toni Kukkohovi

SISÄLLYS

SANASTO	8
1 JOHDANTO	9
1.1 Työn lähtökohta	9
1.2 Päämäärä	11
2 TUTKIMUSPÄIVÄKIRJAN VAATIMUKSET JA OMINAISUUDET	13
2.1 Mobiilisovelluksen vaatimukset	13
2.2 Tutkimuksen käyttötarkoitus	13
2.3 Olemassa olevat tutkimukseen soveltuvat sovellukset	14
2.4 Tutkimuksen vaiheet	15
3 MOBIILISOVELLUKSEN OMINAISUUDET	16
3.1 Tutkimukseen osallistuminen	16
3.2 Ilmoitukset	17
3.3 Kyselyihin vastaaminen	17
3.4 Sovelluksen käytettävyys	18
3.5 Puheen muuntaminen tekstiksi	18
3.6 Mobiilisovelluksen teknologiat	19
4 JÄRJESTELMÄN RAKENNE	20
4.1 Tutkimuksen perustaminen	20
4.2 Datan tallennus tietokantaan	21
5 MOBIILIKEHITYKSEN TYÖKALUT JA MENETELMÄT	22
5.1 Työkalut	22
5.1.1 Git-versionhallinta	22
5.1.2 Android Studio	22
5.2 Ohjelmistokehitysmenetelmät	22
5.2.1 Kotlin	23
5.2.2 MVVM-malli	23
6 MOBIILISOVELLUKSEN TOIMINTA JA KEHITYS	25
6.1 Tutkimukseen osallistuminen	25
6.2 Peruskysymykset	26
6.3 Alkukysymykset	27
6.4 Sovelluksen yhteys palvelimelle sekä JSON-data	28

6.5 Päiväkirjakysymykset	31
6.6 Merkintöjen lisääminen päiväkirjaan	33
7 JATKOKEHITYS	36
7.1 Käyttäjätunnukset	36
7.2 Datan tallennus tietokantaan	36
7.3 Mobiilisovellus iOS-käyttöjärjestelmälle	37
7.4 Järjestelmän käyttö testauksissa	37
7.5 Järjestelmän käyttöönotto	37
8 YHTEENVETO	38

SANASTO

Android	Ohjelmistopino, suunniteltu erityisesti älypuhelimille.
API	Application programming interface. Ohjelmisto- ja ohjelmointirajapinta.
JSON	JavaScript Object Notation. Tiedonvälityksessä käytetty yksinkertainen tiedostomuoto.
Kotlin	JVM-tavukoodiksi kääntyvä yleiskäyttöinen avoimen lähdekoodin staattisesti tyyhitetty ohjelmointikieli.
MVVM	Model-View-ViewModel -ohjelmistoarkkitehtuurimalli. Erottaa käyttöliittymän ohjelman toimintalogiikasta
Tietokanta	Tietojen tallennukseen käytettävä kokoelma, josta voidaan hakea ja muuttaa tietoja.

1 JOHDANTO

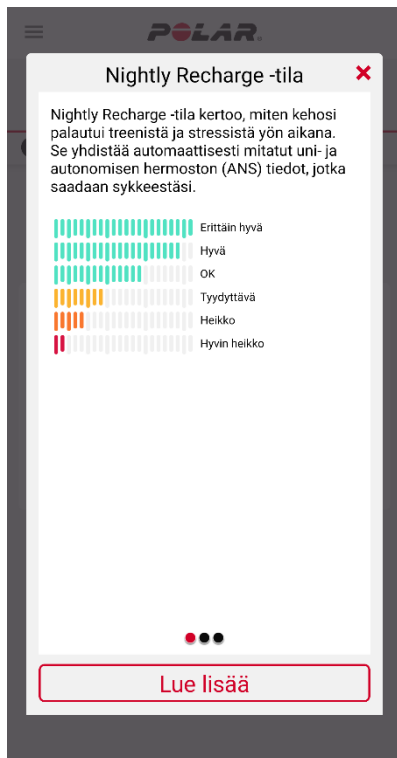
Työn tilaajana toimii tunnettu suomalainen urheiluteknologian kehittäjä ja laitevalmistaja Polar Electro Oy. Polarin pääkonttori sijaitsee Pohjois-Pohjanmaalla Kempeleessä, mutta yrityksellä on myös 26 tytäryhtiötä ympäri maailman. Polar on näyttänyt urheiluteknologian alalla tietä jo vuodesta 1977 asti. (1.)

Polarin tuotteisiin kuuluu sykemittareiden ja urheilukellojen suunnittelu sekä valmistus. Yrityksen toimintaan kuuluu myös nykyisten ja uusien tuotteiden, järjestelmien ja palveluiden kehitys sekä ylläpito.

1.1 Työn lähtökohta

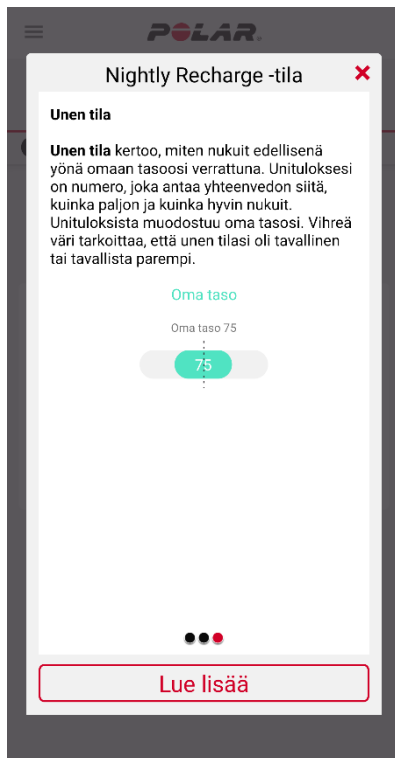
Työlle syntyi idea, kun Polarin tutkimustiimillä ilmeni tarve järjestää erilaisia tutkimuksia ja kerätä dataa tutkimukseen osallistuvien päivän kulusta sekä esimerkiksi urheilijoiden valmistautumisesta erilaisiin urheilutapahtumiin. Päiväkirjatutkimuksessa tutkimukseen osallistuvat henkilöt kirjaisivat mobiilisovellukseen muun muassa, monenko aikaan he menevät nukkumaan, montako kertaa ja milloin he syövät sekä miten ja mitä urheilua he harrastavat päivän aikana. Päiväkirjan tarkemmat kysymykset, kuten ”Oletko stressaantunut tällä hetkellä”, määräytyisivät kuitenkin tutkimuksen mukaan ja ne tulisivat aina tutkimuksen järjestäjältä.

Tutkimuksien avulla Polar pystyy kehittämään erilaisia toimintoja ja lisäominaisuuksia olemassa oleviin sovelluksiin, kuten Polar Flow -mobiilisovellukseen. Polar esimerkiksi kehitti aikaisemmin Nightly Recharge -ominaisuuden, jonka kehityksessä hyödynnettiin unitutkimuksen tuloksia (kuva 1).



KUVA 1. Polar Nightly Recharge -ominaisuus Polar Flow -sovelluksessa

Ominaisuudesta löytyvä ANS-tila kertoo, miten autonominen hermosto on yön aikana rauhoittunut. Tämän lisäksi Nightly Recharge kertoo, kuinka hyvin henkilö on nukkunut sekä palautunut päivän haasteista, kuten urheilusuorituksista ja stressistä (kuva 2).



KUVA 2. Polar Nightly Recharge -ominaisuuden näyttämä unen tila Polar Flow -sovelluksessa

Unitutkimuksessa oli ongelmana tiedon kirjaaminen, sillä tutkimuksen kysymyksiin vastattiin kirjaamalla tulokset käsin paperille tai kännykkään muistiinpanoina. Tämän takia tutkimuksen perustamisesta pitäisi saada mahdollisimman yksinkertainen ja tutkimukseen osallistuvan raportointi eli päivän tapahtumien merkintä mobiilisovellukseen ei saisi olla vaikeaa eikä aikaa vievää. Tutkimuksia voitaisiin myös järjestää esimerkiksi jonkin uuden ominaisuuden testauksessa, jolloin osallistujien olisi helppo raportoida käyttökokemuksista ja ominaisuuden käyttöön liittyvistä ongelmista.

1.2 Päämäärä

Opinnäytetyön päämääränä oli selvittää ja suunnitella tutkimuspäiväkirjan sovelusympäristön rakenne, arkkitehtuuri ja ominaisuudet. Näiden lisäksi tavoitteena oli rakentaa Android-pohjainen mobiilisovellus demo järjestelmän käytöstä. Mobiilisovelluksen vaatimuksena oli perusominaisuudet sekä datan siirtyminen mobiilisovelluksen ja palvelimen välillä. Mobiilisovelluksen avulla voidaan hakea eri näkökulmia käyttöliittymän toteutukselle ja käytettävyydelle. Mobiilisovelluksessa

käyttäjä voi osallistua tutkimukseen Polarilta saamallaan koodilla, täyttää omat tiedot sovellukseen ja vastata tutkimuskohtaisiin alkukysymyksiin. Käyttäjä saa mobiilisovellukselta myös päivittäisistä kyselyistä ilmoituksen, jolloin hänen odotetaan vastaavan kyselyyn. Tämän lisäksi käyttäjä voi lisätä merkintöjä päiväkirjaan. Kaikki käyttäjän lisäämä tieto lähetetään Polarin palvelimelle, josta tutkijat voivat tarkastella ja ladata tietoja analyysia varten.

2 TUTKIMUSPÄIVÄKIRJAN VAATIMUKSET JA OMINAISUUDET

Tutkimuspäiväkirjan tarkoituksena on selvittää, mitä tutkimukseen osallistuva henkilö on päivän aikana tehnyt, ja hyödyntää kerättäviä tuloksia uusien ominaisuuksien kehityksessä. Kaikkea tietoa siitä, mitä henkilö on päivän aikana tehnyt, ei kuitenkaan välttämättä tarvita. Tutkimuspäiväkirjassa tutkimuksen järjestäjä laatii itse oleelliset kysymykset, jotka ovat tutkimuksen kannalta merkittäviä. Tutkimuksen järjestämisen pitää olla helppoa ja siihen osallistumisen yksinkertaista. Tutkimuksella ei saa vaivata siihen osallistuvia henkilöitä eikä kysymyksiin vastaaminen saa olla vaikeaa eikä aikaa vievää. Tämän lisäksi tutkijoiden kannalta analyysin tekemisen täytyy olla helppoa. Tutkimuksissa digitaaliseksi karttuva aineisto nopeuttaa huomattavasti analyysin tekemistä.

2.1 Mobiilisovelluksen vaatimukset

Tutkimukseen osallistuvat henkilöt täyttävät oman aikataulun mukaan tai esimerkiksi illalla päivän tapahtumat, riippuen tutkimuksesta. Jos tutkimuksessa on oleellista nukkumisen seuranta, kyselyyn vastataan esimerkiksi aamulla heti heräämisen jälkeen tai vaihtoehtoisesti illalla ennen nukkumaanmenoa. Reaaliaikaista kyselyn täyttöä varten mobiilisovellukseen täytyy tulla ilmoitus aina, kun kyselyyn pitää vastata. Tutkittaessa stressitasoa tai tunnetilaa on tutkimuksen kannalta suotavaa, että kysymyksiin vastataan heti ilmoituksen saapuessa. Henkilöillä olisi mahdollisuus kirjata päiväkirjatutkimuksessa mobiilisovelluksen avulla päivän tapahtumia käyttämällä sovelluksen hakemia tageja. Tutkijat määrittelevät tutkimuskohtaiset tagit, joita voisivat olla esimerkiksi sauna, rentoutuminen, venyttely, kuuma, kylmä, kipu, koti, väsynyt. Tagien avulla käyttäjän on helppo lisätä päiväkirjamerkintöjä sovellukseen, sillä tagin hakee nopeasti ja aikaleima tulee automaattisesti.

2.2 Tutkimuksen käyttötarkoitus

Tutkimuksen avulla voidaan suunnitella ja kehittää uusia toimintoja ja ominaisuuksia Polarin olemassa oleviin tuotteisiin, kuten rannelaitteisiin ja Polar Flow-mobiilisovellukseen sekä myös tulevaisuudessa kehitettäviin uusiin tuotteisiin. Tutkimuksien avulla voidaan kerätä helposti paljon dataa, jonka avulla voidaan

siten varmistaa erilaisten algoritmien toimintaa ja näin saada kehitettyä uusia hyödyllisiä lisäominaisuuksia ja tuotteita. Tutkimuksiin tarvitaan aina suostumus tutkittavilta, mutta tutkimustuloksia ei kuitenkaan ole tarkoitus kohdistaa henkilöihin. Toisin sanoen tutkittavien henkilötietoja ei tallenneta ja tiedot tallennetaan anonymoineina. Yleensä tutkimustiedot eli tutkimuksista kerättävä data tallennetaan yrityksen suojattuun tietojärjestelmään ja voi olla mahdollista, että tietoja hyödynnetään vielä monen vuoden kuluttua tutkimuksesta.

2.3 Olemassa olevat tutkimukseen soveltuvat sovellukset

Polarilla oli aikaisemmin lyhyt päiväkirjatutkimus, jota varten tutkimuskumppani oli kehittänyt mobiilisovelluksen. Ongelmana tutkimuksessa oli kuitenkin vastausprosentin alhainen kattavuus, sillä kaikki tutkimuksessa mukana olleet henkilöt eivät muistaneet vastata kysymyksiin. Lisäksi toinen ongelma oli se, että moni ei ehtinyt juuri sillä hetkellä vastata kysymyksiin, koska vastausaikaa oli yleensä ainoastaan 30 minuuttia.

Polarilla on myös aiemmin tehty erilaisia tutkimuksia normaalin kirjaamistavan perusteella. Tutkimukseen osallistuville on yleensä annettu kysymykset, joihin osallistuvien odotettiin vastaavan joka päivä. Vastaukset kirjattiin puhelimitse käyttäen Google Keepiä, Microsoft OneNotea tai muuta vastaavaa muistiinpanosovellusta ja palautettiin tutkimuksen järjestäjälle tutkimuksen päätyttyä. Toki aikaisemmissa tutkimuksissa muutamat ovat tyytyneet perinteiseen kirjaustapaan eli kynään ja paperiin.

On olemassa paljon erilaisia työkaluja, joilla voidaan järjestää yksittäisiä kyselyitä. Nämä eivät kuitenkaan sovellu Polarin käyttötarkoituksiin, joissa henkilöiltä vaaditaan usein pitkäaikaista sitoutumista tutkimukseen. Internetistä löytyvillä työkaluilla ja palveluilla tehtävät kyselyt ovat yleensä kertaluontoisia, joten kerättävää dataa jouduttaisiin siten manuaalisesti aina kohdistamaan samaan henkilöön. Lisäksi jos kyseessä on isompi joukko vastaajia, ovat jotkut tällaiset työkalut usein maksullisia. Google Forms on yksi suosituimmista kyselytyökaluista, jolla voidaan tehdä helposti erilaisia kyselyitä, mutta tutkimuksiin, joissa kerätään dataa samoilta henkilöiltä, se ei oikein sovellu. Muita tutkimuspäiväkirjaan soveltuvia sovelluksia ei ole saatavilla, joten Polarilla olisi hyvä olla omaa käyttöä varten

tällainen sovellus. Yrityksen oma sovellus varmistaisi myös tietoturvallisuuden ja henkilötietojen salauksen sekä olisi kustannustehokkain ratkaisu.

2.4 Tutkimuksen vaiheet

Aikaisemmissa tutkimuksissa Polarilla on ollut tapana noudattaa tiettyä järjestystä, jossa on selostettu tutkimuksen eri vaiheet. Tutkimuksen eteneminen perustuu tutkimussuunnitelmaan, joka laaditaan ennen tutkimuksen alkua. Tiedonkeräysvaihe on yleensä aloitettu tutkittavien kutsumisella ja valinnalla. Tutkittavien valintaa varten on määritetty sisäänotto- ja poissulkukriteerit. Tutkimukseen pääsyyn saattavat vaikuttaa erilaiset sairaudet tai se, onko tutkimus tarkoitettu ainoastaan esimerkiksi 30–40-vuotiaille miehille. Alkukyselyt aloitetaan, kun tutkittavia on tiedotettu tutkimuksesta ja heiltä on pyydetty suostumukset tutkimukseen. Tutkittavilta kerätään yleensä alkukyselyssä tutkimuksen tyypin ja laajuuden mukaan perustiedot, kuten ikä, paino, pituus sekä laajemmissa alkukyselyissä lisätietoja muun muassa tutkittavan aktiivisuudesta, harjoittelutaustasta ja sykearvoista. Tutkimuksen alkuvaiheessa saatetaan myös järjestää mittauksia esimerkiksi tutkittavien fyysisestä suorituskyvystä.

Joskus tutkimuksissa saattaa olla tavoitteena muuttaa tutkittavien käyttäytymistä, kuten esimerkiksi treeniä tai nukkumista. Tämän tyyppisissä tutkimuksissa alkutilanteen kartoitus saattaa kestää jopa viikon. Joissain tutkimuksissa tutkittavilta saatetaan kerätä myös rannelaitteella mitattua dataa, esimerkiksi sykemittauksia, nukkumisaikoja ja urheilutuloksia. Näiden takia tutkittavan täyttämässä tutkimuspäiväkirjassa eli päivittäisissä kysymyksissä aikaleima on tärkeä, jotta dataa voidaan yhdistää.

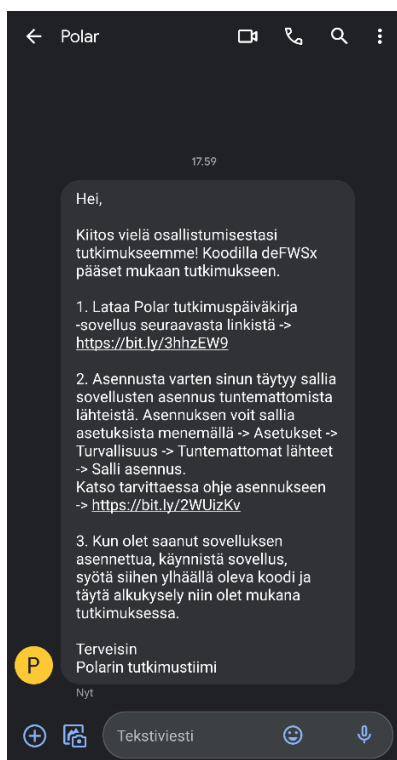
Yleensä tutkimus voi alkaa tietyssä aikana, jolloin kaikkia siihen osallistuvia henkilöitä seurataan jonkin ennalta määrätyn ajan. Tutkimus voi myös alkaa liukuvasti eli niin, että siihen voi tulla mukaan kesken tutkimuksen. Liukuvalla ajalla alkavat tutkimukset ovat yleensä kestoiltaan pitempiä tutkimuksia. Tällaisen tutkimuksen seuranta-aika voi kestää kuukausia tai jopa vuosia. Tutkimuksen päätyttyä saatetaan tehdä loppumittaukset tutkittaville, kuten painon ja suorituskyvyn mittaus uudelleen. Näin voidaan selvittää, onko tutkimuksella saatu aikaiseksi muutosta alkutilanteeseen nähden.

3 MOBIILISOVELLUKSEN OMINAISUUDET

Mobiilisovelluksen kehityksessä on tärkeää ottaa huomioon mobiilisovelluksen helppokäyttöisyys kuluttajien tai tässä tapauksessa tutkittavien kannalta. Sovelluksen täytyy olla helposti käyttöönotettava, helppokäyttöinen sekä yksinkertainen. Mobiilisovelluksen voisi ladata pakettina suoraan puhelimeen esimerkiksi Polarin tarjoamasta linkistä. Sovellus ei saa olla toiminnaltaan hidas eikä se saa jumittaa missään tilanteessa. Nykyaikaiset mobiilisovellukset kaatuvat todella harvoin, sillä niissä on yleensä toimiva virnehallinta. Tutkimukseen käytettävässä mobiilisovelluksessa on tärkeää ottaa huomioon datan tallennus ja lähetys, jos puhelin ei ole yhteydessä verkkoon sillä hetkellä, kun käyttäjä vastaa kyselyyn.

3.1 Tutkimukseen osallistuminen

Jokaisella tutkimuksella tulee olla oma yksilöllinen koodi, jonka tutkimuksen järjestäjä antaa tutkimukseen osallistuville henkilöille. Tutkimukseen osallistuvat henkilöt pääsevät osallistumaan tutkimukseen lataamalla mobiilisovelluksen esimerkiksi viestillä saapuvasta linkistä. Android on yksi yleisimmistä monissa eri valmistajan älypuhelimissa toimivista käyttöjärjestelmistä ja iOS-käyttöjärjestelmä on käytössä vain Applen valmistamissa iPhone-älypuhelimissa. Androidilla ja iOSilla on yleensä estetty ulkopuolisten sovellusten asennus tuntemattomista lähteistä. Tätä varten viestissä täytyy olla osallistumiskoodin lisäksi ohje, miten puhelimesta sallitaan ulkopuolisten sovellusten asennus. Tuntemattomilla lähteillä tarkoitetaan sovelluksia, jotka ladataan muualta kuin valmistajan ylläpitämistä sovelluskaupoista. Androidilla varmistettuja sovelluksia voi ladata vapaasti ainoastaan Google Play -kaupasta ja iOSilla App Storesta. Sovelluksen asennettuaan käyttäjä syöttää osallistumiskoodin mobiilisovellukseen ja on näin mukana kyseisessä tutkimuksessa. Tutkimukseen osallistuville tulisi tutkimuksen alkaessa esimerkiksi kuvan 3 mukainen viesti.



KUVA 3. Esimerkkiviesti tutkimukseen osallistuville

3.2 Ilmoitukset

Yksi tärkeimpiä ominaisuuksia mobiilisovelluksessa on ilmoitukset. Käyttäjälle tulisi ilmoitus heti, kun kyselyyn pitää vastata. Ilmoituksissa eli niin sanotuissa notifikaatioissa olisi pikavalinnat, joista käyttäjä voisi helposti torkuttaa kyselyä avaamatta sovellusta. Ilmoituksessa olisi torkutusmahdollisuus aina puolen tunnin päähän, jolloin ilmoitus kyselystä tulisi uudelleen. Käyttäjällä olisi mahdollisuus torkuttaa ilmoitusta tiettyyn rajaan asti. Kyselyyn vastaamisen raja olisi, joko ennen seuraavaa kyselyä tai päivän loppuun mennessä. Tutkijalla pitää kuitenkin olla mahdollisuus määrittää tarkka raja, milloin kyselyyn pitää viimeistään vastata.

3.3 Kyselyihin vastaaminen

Vaikka tutkimuksessa mukana olevat henkilöt saisivat ilmoituksen kyselyyn vastaamisesta, eivät he välttämättä ole juuri sillä hetkellä puhelimen äärellä tai heillä ei ole esimerkiksi työn puolesta mahdollisuutta tai aikaa vastata kyselyyn. Tämän takia kyselyn aikaraja pitäisi olla pitempi, mutta esimerkiksi kyselyissä, joissa hetkellinen vastaus on tärkeä, voitaisiin ottaa huomioon, kuinka myöhässä henkilö

vastasi kysymykseen. Myöhästyneille vastauksille voitaisiin tutkimuksen kartoituksessa määrittää painoarvo. Esimerkkinä tutkimuksessa heti tai mahdollisimman pian vastauksen vaativia kysymyksiä voisivat olla ”Oletko väsynyt?”, ”Kuinka stressaantunut olet?” ja ”Oletko nälkäinen?”.

3.4 Sovelluksen käytettävyys

Mobiilisovelluksen käytettävyyden täytyy olla helppoa eikä se saa tuottaa vaikeuksia. Kun kyselyihin vastaaminen on yksinkertaista ja tutkimukseen osallistuminen ei vie aikaa, on selvää, että vastaukset ja kerätty data ovat luotettavampia, koska tällöin käyttäjällä on paljon suurempi motivaatio käyttää sovellusta. Sovelluksen käyttäjää motivoisi varmasti lisää se, jos sovelluksessa olisi esimerkiksi nappi, jota painamalla vastaukset lähetettäisiin Polarin tietokantaan ja käyttäjä saisi tästä ilmoituksen, kuten ”Vastauksesi on lähetetty”. Tällöin käyttäjälle tulisi kyselyihin vastatessa tunne, että hänen vastauksillaan on oikeasti merkitystä. Tällaisten pienten asioiden avulla saadaan käyttäjät vastaamaan kyselyihin ajallaan.

Sovelluksen ulkoasussa tärkeää on sen yksinkertaisuus, tekstien fontti, fonttikoko sekä värit. Jos mobiilisovelluksen käytettävyys on sujuvaa ja ongelmattonta, saadaan sovelluksen käyttäjät raportoimaan aktiivisesti tuloksia tutkimukseen. Sovelluksen ulkoasu on sitä yksinkertaisempi, mitä vähemmän sovelluksessa on ominaisuuksia. Juuri tämän takia sovelluskehityksessä pitää miettiä tarkkaan läpi jokaisen ominaisuuden kohdalla, onko ominaisuus oikeasti hyödyllinen käyttäjän kannalta vai onko se vain turha lisäominaisuus, jolle ei välttämättä ole käyttöä.

3.5 Puheen muuntaminen tekstiksi

Yksi sovelluksen hyödyllisistä ominaisuuksista olisi puheen muuntaminen tekstiksi eli kysymyksiin vastaaminen puhumalla. Kysymyksiin, jotka vaativat lisätietoa tai kirjoittamista käyttäjältä, voisi olla helpompaa vastata puhumalla. Käyttäjä painaisi vain mikrofonin nappia, jolloin hän voisi puhua vastauksensa kysymykseen. Tällainen käyttötapa voisi olla esimerkiksi silloin, jos käyttäjä kertoisi nukkuneensa huonosti, jolloin hän voisi antaa lisätietoa vastaamalla puhumalla kysymykseen ”Miksi nukuit huonosti”. Google tarjoaa jo omaa ratkaisua tähän, sillä

Googlen G-Board-näppäimistössä on ominaisuutena puhekirjoitus. Tällä puhekirjoituksella tarvitsee painaa vain mikrofonicuvaketta ja puhua, mitä halutaan kirjoittaa.

3.6 Mobiilisovelluksen teknologiat

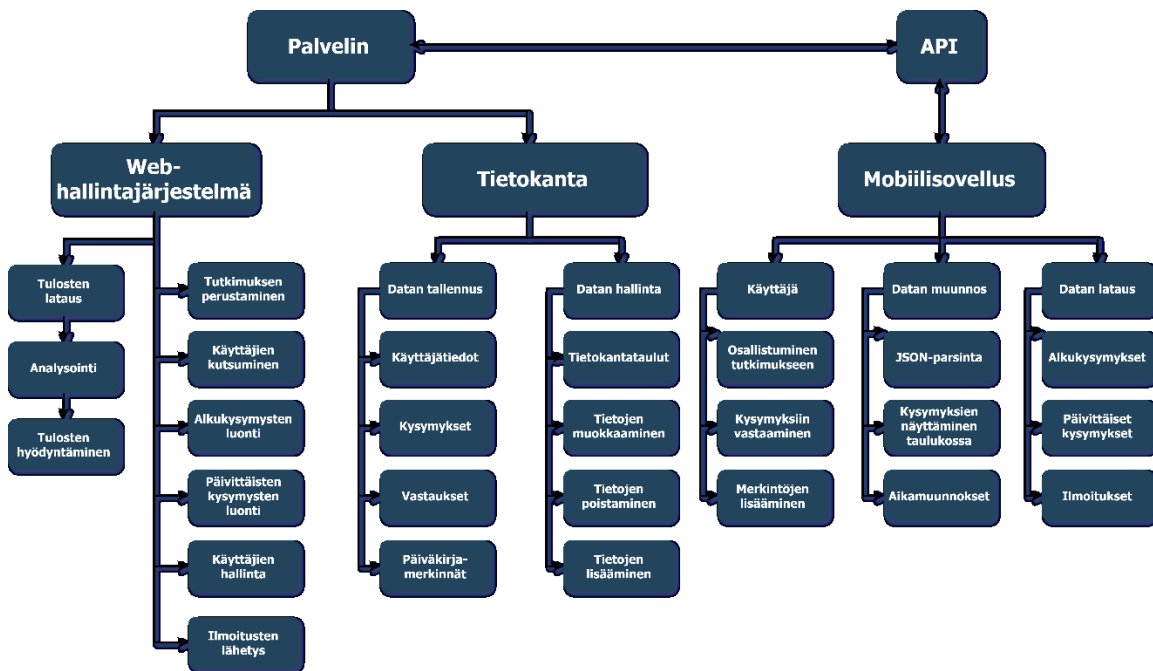
On hyvin yleistä, että mobiilisovellus tehdään natiivisovelluksena eli erikseen Androidille ja iOSille. Androidille sovellus yleensä toteutetaan käyttämällä Java- tai Kotlin-ohjelmointikieltä ja iOSilla on yleensä käytetty Applen omaa Swift-kieltä, joka on hyvin paljon samankaltainen Objective C -kielen kanssa. Mobiilisovellus olisi kuitenkin mahdollista tehdä käyttäen cross-platform-teknologiaa, esimerkiksi React Native -teknologialla, jolloin tarvitsee kehittää vain yksi sovellus, joka sitten käännettäisiin sekä Androidille että iOSille. Yleisesti natiivisovellukset ovat cross-platform -teknologiaa suositumpia, sillä natiivisovellusten etuja ovat niiden suorituskyky, tietoturvallisuus sekä helpompi pääsy laitteen eri komponentteihin, kuten sijaintitietoon ja kameraan (taulukko 1). Cross-platform -kehityksessä työmäärä on vähäisempi ja kehitys nopeampaa, joten se sopisi sellaiseen sovelluskehitykseen, jossa aikataulu on kiireellinen ja mobiilisovellukselta ei vaadita kovin monipuolisia ominaisuuksia. (2.)

TAULUKKO 1. Natiivin ja cross-platform-teknologioiden eroavaisuudet (1).

	Natiivi	Cross-platform
+	Turvallisuus Energiansäästö Sovellus vakaa ja nopea Mahdollisuus käyttää uusia teknologioita Tulevaisuuden kehityskustannukset	Nopea markkinointi Nopea kehitys Kustannukset
-	Pääasialliset kustannukset Kehitykseen käytetty aika	Ohjelmistokehittäjien puute Versioiden väliset ongelmat
=	Sovellusten kehitykseen Vaikeille ja vaativille toiminnoille Puhelimen omien toimintojen käyttö Offline ja online	Yksinkertaisille sovelluksille Prototyypeille ja markkinointitestaukselle Ei tarvetta puhelimen toiminnoille

4 JÄRJESTELMÄN RAKENNE

Järjestelmäkokonaisuuden kehittämisessä on tärkeää ottaa huomioon sen rakenne eli järjestelmäarkkitehtuuri. Ohjelmistoaikalla on myös paljon erilaisia arkkitehtuurimalleja, joten sekin on hyvä ottaa huomioon mobiilisovellusta rakennettaessa. Tässä työssä datan tallennukseen käytetään Polarin omaa tietokantaa, josta data haetaan puhelimeen JSON-muodossa ja muunnetaan mobiilisovelluksessa käyttäjälle näytettävään graafiseen muotoon. Tulevaisuudessa kehitettävä järjestelmä rakentuu pääasiassa Polarin omilla palvelimella toimivasta tietokannasta, mobiilisovelluksesta ja selaimessa toimivasta tutkijoille tarkoitetusta hallintajärjestelmästä. (Kuva 4.)



KUVA 4. Tutkimuspäiväkirjan järjestelmän rakennekaavio

4.1 Tutkimuksen perustaminen

Tutkijalla tulisi olla tulevaisuudessa käytössään selaimessa toimiva käyttöliittymä, jonka avulla hän voi luoda tutkimuksia. Tutkijoille voisi myös luoda omat käyttäjätunnukset, joilla he pääsisivät kirjautumaan palveluun sisään. Tutkimuksen perustaja voisi myös selainpohjaisen käyttöliittymän avulla lisätä tutkimukseen mu-

kaan muita henkilöitä hallitsijoiksi. Muut henkilöt voisivat tarkastella tutkimuksessa kerättyä dataa ja muuttaa tarvittaessa muita tutkimuksen luoneen henkilön tekemiä asetuksia ja määrittämiä. Tutkimuksen luonnin jälkeen tutkija määrittelee tarvittaessa alkukysymykset, jotka kysytään ennen varsinaista tutkimusta tutkimukseen osallistuvilta henkilöiltä, ja näiden lisäksi sitten toistuvat kysymykset tutkimukselle. Tämän jälkeen tutkimukselle generoidaan automaattisesti yksilöllinen tunnistekoodi eli ID, jotta tutkimukseen voidaan kohdistaa oikeaa dataa. Tutkimuskohtaisen tunnistekoodin lisäksi tutkija generoi jokaiselle käyttäjälle yksilöllisen osallistumiskoodin, jonka avulla käyttäjät voivat osallistua tutkimukseen.

Tietokannasta löytyy lista kaikista käyttäjistä, jotka ovat mukana tutkimuksessa. Listassa on ainoastaan käyttäjien ID:t, sillä nimiä ei tallenneta järjestelmään ollenkaan. Tämän listan avulla voidaan lähettää ilmoitukset tutkimuksen kysymyksistä jokaiselle käyttäjälle samaan ja oikeaan kellonaikaan. Tutkija päättäisi, miten usein ja milloin kysymykset tulevat tutkittaville.

4.2 Datan tallennus tietokantaan

Kun käyttäjä syöttää saamansa osallistumiskoodin mobiilisovellukseen, sovellus lähettää tietokantaan pyynnön ja saa siten tietokannasta vastineeksi tunnistekoodin eli ID:n, joka tallennetaan myös puhelimeen. Tätä ID:tä käytetään datan tallentamisessa oikeaan paikkaan aina, kun puhelin lähettää dataa tietokantaan. Jokaisella tutkittavalla on siis oma ID, jotta data kohdistuu aina oikeaan paikkaan. Kun käyttäjä vastaa esimerkiksi päivän aikana tuleviin kysymyksiin ja painaa tallennuspainiketta, lähetetään vastaukset tietokantaan oikeaan datatauluun käyttämällä aiemmin generoitua ID:tä. Tietokannasta haetaan myös kaikki data sovellukseen yhdistelemällä tarvittavia tutkimuksen, käyttäjän ja kysymyksiä ID-tunnisteita. Käyttäjän vastatessa päivittäisiin kysymyksiin tai lisätessä esimerkiksi merkintöjä päiväkirjaan tallennetaan samalla kertaa aikaleima varsinaisen vastauksen perään.

5 MOBIILIKEHITYKSEN TYÖKALUT JA MENETELMÄT

Tässä luvussa kerrotaan tämän työn mobiilisovelluksen kehityksessä käytettävistä työkaluista ja menetelmistä.

5.1 Työkalut

Mobiilisovelluksen kehityksessä versionhallintaan käytetään Polarin sisäistä Git-versionhallintaa sekä itse kehitykseen Googlen Android Studio -ohjelmointityökalua.

5.1.1 Git-versionhallinta

Mobiilisovelluksen kehityksessä koodin hallintaan käytetään Polarin sisäistä Git-versionhallintajärjestelmää. Git on vuonna 2005 Linus Torvaldsin julkaisema hajautettu versionhallintaan käytettävä työkalu (6). Git helpottaa huomattavasti ohjelmistokehitystyötä, sillä sen avulla voidaan muokata ja yhdistää tiedostoja, kuten toimintoja, joita lisätään sovellukseen ohjelmistokehityksen eri vaiheissa. Gitin avulla voidaan myös ongelmien syntyessä helposti palata sovelluksen edellisiin versioihin sekä ohjelmistokehittäjät pysyvät myös helposti mukana toisten kirjoittamissa koodeissa Gitin ansiosta.

5.1.2 Android Studio

Suosituin ohjelmointiympäristö Android-pohjaisten sovellusten kehityksessä on Googlen ja JetBrainsin kehittämä, vuonna 2014 julkaistu Android Studio. Android Studiosta löytyvät Intellij IDEA -ohjelmointiympäristön lisäksi kattavat ja tehokkaat ominaisuudet mobiilisovelluskehitykseen, kuten emulaattori, testausympäristö, tarvittavat rajapinnat ja Googlen Firebase -tietokantatuki. (7.)

5.2 Ohjelmistokehitysmenetelmät

Uuden ohjelmistoprojektin aloituksessa olisi aina suositeltavaa käyttää uusimpia ja tehokkaimpia teknologioita sekä työkaluja, koska ne tuovat sovellukselle lisää käyttöikää sekä helpottavat sen rakentamista. Mobiilisovelluksen kehityksessä

on tänä päivänä suositeltavaa käyttää Kotlin-kieltä Java-kielen sijaan sekä arkkitehtuurimallina olisi hyvä olla yksi suosituimmista malleista, joka on MVVM-malli (7).

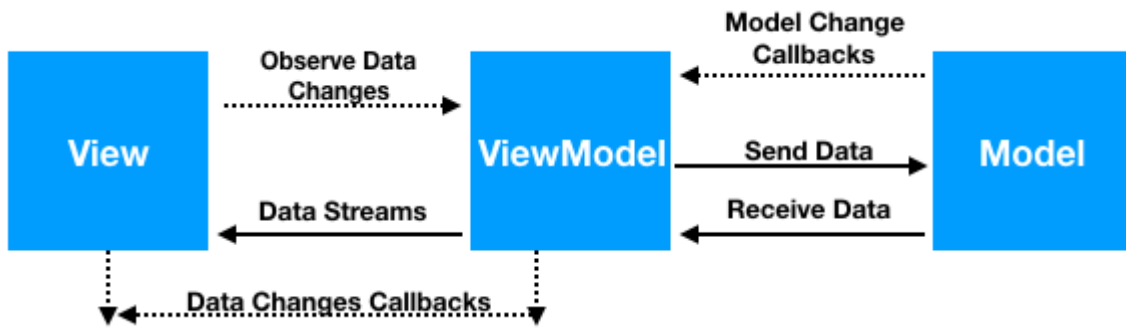
5.2.1 Kotlin

Kotlin on Googlen hyväksymä virallinen Android-sovellusten ohjelmointikieli. Google on virallisesti käyttänyt ja suositellut mobiilikehitykseen aiemmin Java-kieltä, kunnes Kotlin julkaistiin vuonna 2016 ja Google ilmoitti vuonna 2019 Kotlinin olevan Android-sovelluskehityksen suositeltu kieli. Kotlin kehitettiin Javan korvaajaksi Android-kehitykseen, sillä se, mihin tarvitaan 50 riviä Java-koodia, voidaan kirjoittaa muutamalla rivillä Kotlinia. Usein Kotlinia on kuvailtu paremmaksi kuin Java, vaikka se onkin Javan kanssa täysin yhteensopiva. (3.) Tässä projektissa käytetään Kotlin-kieltä Android-sovelluksen kehityksessä.

5.2.2 MVVM-malli

Android-pohjaisten mobiilisovellusten käytössä on ollut jo jonkin aikaa nopeasti suosiota kasvattanut MVVM-ohjelmistoarkkitehtuurimalli. Mobiilisovelluskehityksessä, kuten muissakin ohjelmistokehityksissä, on tärkeää valita oikea arkkitehtuurimalli, joka helpottaa ohjelmointityötä huomattavasti. Ilman arkkitehtuurimalia sovelluksen ja koodin hallinta menee sitä sekavamman, mitä enemmän koodia tulee. Tässä projektissa käytetään MVVM-mallia, sillä Google on antanut suosituksensa MVVM-ohjelmistoarkkitehtuurin käytölle Android-ohjelmoinnissa (4).

MVVM-arkkitehtuurimalli on lyhennetty englanninkielisestä nimestä Model-View-ViewModel, joka kääntyy suomeksi muotoon Malli-Näkymä-Näkymämalli. MVVM-malli pyrkii erottamaan graafisen käyttöliittymän itse sovelluksen toiminnasta ja datalogiikasta, jonka avulla se varmistaa sen, että mobiilisovelluksen näkymä eli graafiset komponentit eivät ole riippuvaisia itse mallista ja datasta (kuva 5). ViewModel (näkömämalli) toimittaa Modelilta (mallilta) kaikki tarvittavat päivitykset View-osalle (näkömälle), joka hoitaa datan näyttämisen ja päivittämisen käyttäjälle. ViewModel toimii myös toisinpäin, sillä se hoitaa esimerkiksi käyttäjän integraatiot sovelluksessa ja toimittaa ne View'ltä Modelille (näkömältä mallille). (4.)



KUVA 5. MVVM-mallin toiminnan kuvaus (5)

6 MOBIILISOVELLUKSEN TOIMINTA JA KEHITYS

Mobiilisovelluksen kehityksessä tarkoituksena ei ollut rakentaa täysin toimivaa, valmista applikaatiota, vaan ainoastaan antaa alustavan sovelluksen avulla yleiskuva järjestelmän toiminnasta sekä hakea käyttöliittymän ulkonäön avulla eri näkökulmia sovelluksen toteutukselle. Aloitin mobiilisovelluksen kehityksen luomalla Git-versionhallintajärjestelmään uuden projektin. Versionhallintajärjestelmässä päähakemistona toimi Develop-hakemisto, johon lisättiin sitä mukaan ominaisuuksia, kun niitä rakennettiin. Kehitysympäristönä projektissa toimi Googlen Android Studio ja ohjelmointikielenä Kotlin.

6.1 Tutkimukseen osallistuminen

Käyttäjän käynnistäessä sovelluksen ensimmäistä kertaa näytetään tervetuloa näkymä (kuva 6). Näkymässä käyttäjä syöttää sovellukseen osallistumiskoodin, joka lähetetään palvelimelle tarkistettavaksi. Jos käyttäjä ei ole jostain syystä saanut koodia tai on esimerkiksi hukannut koodin, näkymän alareunassa on linkki ”Minulla ei ole koodia”, joka tulevaisuudessa tulee ohjaamaan käyttäjän sivulle, josta hän voi ottaa helposti yhteyttä tutkimuksen järjestäjään.



Tervetuloa Polarin tutkimukseen

Kiitos kun haluat tulla mukaan tutkimukseemme. Me Polarilla kehitämme tulevaisuuden urheiluteknologiaa ja sinä voit olla mukana auttamassa meitä saavuttamaan tavoitteemme. Syötä saamasi osallistumiskoodi ja liity mukaan tutkimukseen.

Kirjoita koodi tähän

LIITY

Minulla ei ole koodia

KUVA 6. Tervetuloa näkymä mobiilisovelluksessa

Osallistumiskoodin tarkistuksessa ViewModel saa View'ltä osallistumiskoodin ja käynnistää Coroutines-pohjaisen prosessin, joka ottaa API:n avulla yhteyden palvelimelle ja saa oikealla osallistumiskoodilla käyttäjälle yksilöllisen ID:n sekä tutkimuskohtaisen ID-tunnuksen. Nämä ID:t tallennetaan puhelimeen ja niitä käytetään aina, kun kommunikoidaan palvelimen kanssa. Näiden lisäksi puhelimeen tallennetaan tieto siitä, että käyttäjä on osallistunut tutkimukseen. (Kuva 7.)

```
fun checkParticipationCode(participationCode: String) {
    networkListener?.onStarted()
    Coroutines.main {
        try {
            val loadResponse : Participate = repository.checkParticipationCode(participationCode)
            loadResponse.let { it: Participate
                if (it.status == 1) {
                    networkListener?.onSuccess("")
                    sharedPreferences.saveResearchSubscription()
                    sharedPreferences.saveUserID(it.userId)
                    sharedPreferences.saveResearchID(it.researchId)
                } else {
                    networkListener?.onFailure(errorMessage)
                    return@main
                }
            }
        } catch (e: NoInternetException) {
            networkListener?.onFailure(e.message!!)
        }
    }
}
```

KUVA 7. Osallistumiskoodin tarkistus palvelimelta

Jos osallistumiskoodi on oikea, tehdään tarvittavat, käyttäjälle näkymättömät toimenpiteet ja siirrytään seuraavaan näkymään, jossa kysytään käyttäjältä peruskysymykset.

6.2 Peruskysymykset

Tutkimuksien alussa on peruskysymykset, jotka kysytään jokaiselta käyttäjältä, jokaisessa tutkimuksessa. Tähän saattaa kuitenkin tulla tulevaisuudessa muutos. Aika monessa tutkimuksessa on kuitenkin yleistä, että tutkimuksen alkukysymyksissä kysytään osallistujien syntymäaika, pituus, paino sekä sukupuoli. (Kuva 8.)



Perustiedot

Syntymäpäivä 20.10.1997

Pituus cm

Paino kg

Sukupuoli Mies
 Nainen

JATKA

KUVA 8. Tutkimukseen osallistujan perustiedot

Käyttäjän täyttäessä perustiedot ja painaessa Jatka-painiketta, lähetetään kysymykset POST-kyselyllä, API:n avustuksella palvelimelle ja haetaan samalla yhteydenmuodostuksella tutkimuksen alkukysymykset.

6.3 Alkukysymykset

Perustietojen jälkeen käyttäjän täytyy vastata palvelimelta haettaviin, tutkijan määrittelemiin alkukysymyksiin (kuva 9). Alkukysymykset haetaan taulukkona ja niiden tyyppi tarkistetaan ennen niiden esittämistä käyttäjälle. Kysymyksiä voi olla kolmea eri tyyppiä. Monivalintakysymyksissä käyttäjä voi valita usean vaihtoehdon, peruskysymyksissä on kaksi vaihtoehtoa, joista käyttäjä valitsee yhden, ja kolmantena on liukusäädintyyppinen kysymys, jossa käyttäjä valitsee liukusäätimen avulla sopivan arvon esimerkiksi yhden ja kymmenen väliltä. Asteikko, josta käyttäjä valitsee sopivan arvon, on tutkijan määrittelemä sekä kysymyskohtainen. Jokaiseen kysymykseen on myös määriteltä, onko kysymykseen pakko vastata. Jos kysymys ei ole pakollinen, käyttäjä voi ohittaa sen helposti painamalla kysymyskortin oikeassa yläreunassa olevaa x-painiketta. Käyttäjän osallistuminen tutkimukseen on tallennettu puhelimeen, joten vaikka käyttäjä poistuisi alkukysy-

myksien kohdalla sovelluksesta, ei hänen tarvitse seuraavalla kerralla enää syöttää osallistumiskoodia sovellukseen eikä täyttää perustietoja uudelleen, vaan hän voi jatkaa alkukysymysten täyttämistä.

< **POLAR**

Alkukysymykset 2/6

Hyvinvointi, työn-imukysely (Work engagement)

Kuinka usein sinulla on seuraavien väittämien kaltaisia tunteuksia tai ajatuksia? 1 = Ei koskaan, 2 = Muutaman kerran vuodessa, 3 = Kerran kuussa, 4 = Muutaman kerran kuussa, 5 = Kerran viikossa, 6 = Muutaman kerran viikossa, 7 = Päivittäin

Tunnen olevani täynnä energiaa kun teen työtäni

1 7

Tunnen itseni vahvaksi ja tarmokkaaksi työssäni

1 7

Olen innostunut työstäni ✘

SEURAAVA

KUVA 9. Tutkimuksen alkukysymykset

6.4 Sovelluksen yhteys palvelimelle sekä JSON-data

Palvelimena projektissa toimii Polarin oma palvelin, jonka Polarin työntekijä laittoi toimimaan. Dataa haetaan palvelimelta käyttäen Retrofit-kirjastoa (kuva 10). Retrofit on Squaren kehittämä REST-tyyppinen asiakasrajapinta, joka on kehitetty Javalle ja Kotlinille. REST tulee sanoista Representational State Transfer ja se on arkkitehtuurimalli, joka perustuu HTTP-protokollaan. Retrofit tarjoaa tehokkaan kehiksen ja rajapinnan käyttäjän todennukseen, verkkopyyntöihin ja API-yhteyksiin. (8.)

```

companion object {
    operator fun invoke(
        networkConnectionInterceptor: NetworkConnectionInterceptor
    ): API {
        val client : OkHttpClient = OkHttpClient.Builder()
            .addInterceptor(networkConnectionInterceptor)
            .connectTimeout(TIME_OUT, TimeUnit.SECONDS)
            .readTimeout(TIME_OUT, TimeUnit.SECONDS)
            .writeTimeout(TIME_OUT, TimeUnit.SECONDS)
            .build()

        return Retrofit.Builder()
            .baseUrl(CODE_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .client(client)
            .build()
            .create(API::class.java)
    }
}

```

KUVA 10. Retrofit-kirjaston käyttö datan haussa

Yhteyden muodostamisfunktiolle "Invoke" annetaan parametrinä NetworkConnectionInterceptor-luokka, joka varmistaa ennen yhteyden muodostamista, että puhelin on yhteydessä verkkoon. Sovellus hakee dataa palvelimelta JSON-muodossa, joka muunnetaan sovelluksessa helposti käytettävään muotoon. Datan muunnoksessa hyödynnetään niin sanottuja model-luokkia. Model-luokista QuestionItem- ja QuestionInfo-luokkien avulla tietoja voidaan hakea helposti. Luokkien avulla voidaan viitata listan oikeaan indeksiin ja hakea siitä kohdasta nimellä esimerkiksi kysymyksen tiedot ja vastausvaihtoehdot.

JSON tulee sanoista JavaScript Object Notation, ja vaikka se viittaa JavaScript-kieleen, se on kuitenkin täysin riippumaton JavaScriptistä. JSON-muotoinen data on ihmiselle luettavaa ja dataa voidaan käyttää useiden eri ohjelmointikielien kanssa. (9.) Alkukysymyksen mukana saadaan aiheotsikot, kysymysten lisätiedot, vastausvaihtoehdot sekä muut tarpeelliset tiedot helposti (kuva 11).

```

▼ 1:
  heading: "Hyvinvointi, työn-imukysely (Work engagement)"
  ▶ headingInfo: "Kuinka usein sinulla on ...ikossa, 7 = Päivittäin "
  answerType: 3
  ▼ questions:
    ▼ 0:
      questionId: 5698
      order: 2
      question: "Tunnen olevani täynnä energiaa kun teen työtäni"
      questionInfo: ""
      mandatory: true
      answerFrom: 1
      answerTo: 7
      ▼ answers:
        0: {}
    ▼ 1:
      questionId: 4
      order: 3
      question: "Tunnen itseni vahvaksi ja tarmokkaaksi työssäni"
      questionInfo: ""
      mandatory: true
      answerFrom: 1
      answerTo: 7
      ▼ answers:
        0: {}
    ▶ 2: {...}
    ▶ 3: {...}

```

KUVA 11. Alkukysymysten JSON-data

Alkukysymyksien datassa heading-kenttä antaa otsikon yhdelle kysymysosiolle ja answerFrom- sekä answerTo-kentät kertovat numerot, joiden väliltä käyttäjä voi valita arvon käyttäen liukusäädintä. Mandatory-kenttä määrittää, onko kysymykseen pakko vastata. Kuvassa 11 JSON-data näytetään selaimen muuntamassa, luettavassa muodossa. Todellisuudessa data on kuitenkin hieman mutkikkaampaa luettavaa (kuva 12).

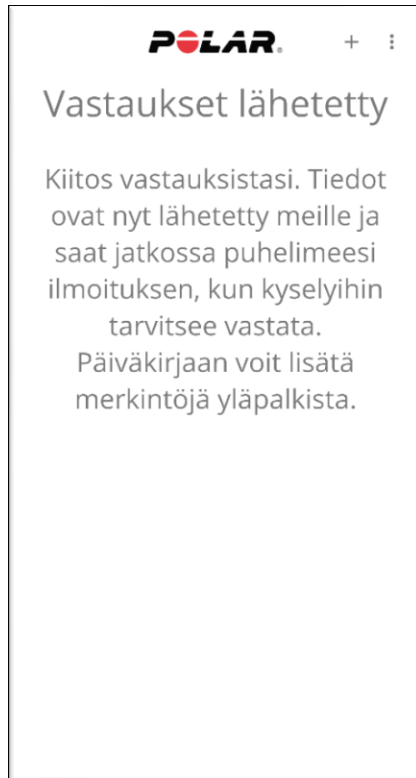
```
[
  {
    "heading": "Terveys ja terveyskäyttäytyminen",
    "headingInfo": "",
    "answerType": 3,
    "questions": [
      {
        "questionId": 1,
        "order": 0,
        "question": "Olettaen, että työkykysi on parhaimmillaan saanut 10 pistettä. Minkä pistemäärän antaisit nykyiselle työkyvyyllesi?",
        "questionInfo": "1 tarkoittaa sitä, ettet nykyisin pysty lainkaan työhön",
        "mandatory": true,
        "answerFrom": 1,
        "answerTo": 10,
        "answers": [
          {}
        ]
      },
      {
        "questionId": 1232,
        "order": 1,
        "question": "Miten arvioit terveydentilasi suhteessa muihin samanikäisiin?",
        "questionInfo": "1 = Erittäin huono, 2 = Huono, 3 = Keskinkertainen, 4 = Hyvä, 5 = Erittäin hyvä",
        "mandatory": false,
        "answerFrom": 1,
        "answerTo": 5,
        "answers": [
          {}
        ]
      }
    ]
  }
],
}
```

KUVA 12. JSON-datan perusmuoto

Sovelluksen päivittäisissä kysymyksissä käytetään samantyylistä JSON-rakennetta sekä kysymykset ladataan käyttäen Retrofit-kirjastoa.

6.5 Päiväkirjakysymykset

Käyttäjän täytettyä alkukysymykset siirrytään näkymään, jossa käyttäjälle näytetään tieto siitä, että kysymykset lähetettiin palvelimelle. Samassa näkymässä palvelimelta ladataan varsinaiset tutkimuksen aikana käyttäjälle tulevat kysymykset. (Kuva 13.)



KUVA 13. Vastaukset lähetetty -näkyvä

Ladattavien kysymysten mukana on tieto siitä, monenko aikaan jokaiseen kysymykseen käyttäjän odotetaan vastaavan. Tähän kellonaikaan lähetetään myös käyttäjälle ilmoitus kyselystä. Käyttäjän saadessa ilmoituksen uudesta kyselystä, sovellus avautuu painamalla saapunutta ilmoitusta ja näyttää avauduttuaan uusimmat kysymykset listassa, joihin käyttäjä voi vastata. Jos käyttäjä ei ole ehtinyt vastata mahdollisiin päivän aikana jo aiemmin ilmestyneisiin kysymyksiin, näytetään kysymykset kronologisessa järjestyksessä, uusimmasta vanhempaan. Käyttäjällä on aikaa vastata päivän kysymyksiin kyseisen päivän loppuun saakka eli kello 00.00 asti. (Kuva 14.)

Tervetuloa

Täytä aamukysely

Monta kertaa heräsit yön aikana? ✕

1 = Yli 5 kertaa, 2 = 3-4 kertaa, 3 = 1-2 kertaa, 4 = En kertaakaan

1 4

Miten hyvin nukuit? ✕

1 = Erittäin huonosti, 2 = Huonosti, 3 = Keskinertaisesti, 4 = Hyvin, 5 = Erittäin hyvin

1 5

TALLENNA

KUVA 14. Päivän aikana vastattava kysely

6.6 Merkintöjen lisääminen päiväkirjaan

Käyttäjä voi lisätä merkintöjä päiväkirjaan yläpalkissa olevasta plusmerkistä. Merkkiä painamalla siirrytään "Lisää merkintä" -näkyymään, jossa käyttäjä voi täyttää tarvittavat tiedot. Kellonaika-kohtaan haetaan automaattisesti nykyinen aika ja päivämäärä, mutta käyttäjä voi muokata aikaa sekä päivämäärää napauttamalla. Muokkaus voi olla tarpeellista, jos käyttäjä on esimerkiksi unohtanut lisätä jonkin merkinnän päiväkirjaan tai haluaa lisätä tulevaisuuden merkinnän. Lisätiedot-kenttään käyttäjä voi halutessaan kirjoittaa lisätietoja merkinnästä. Tallenna-painiketta painamalla siirrytään takaisin edelliseen näkyymään. (Kuva 15.)

< **POLAR**

Lisää merkintä

Tagi

Lisää tapahtuma

Päivämäärä

28. lokakuuta 2020

Kellonaika

20.50

Lisätietoja

VALMIS

KUVA 15. Merkinnän lisääminen päiväkirjaan

Käyttäjän painaessa tagin tekstikenttää avautuu lista, josta voi hakea päiväkirja-merkintää vastaavan tagin (kuva 16).

< **POLAR**

Lisää merkintä

Tagi

Ty

Matka kotoa työpaikalle/työpaikalta..

Työ, alku

Työ, loppu

Lisätietoja

KUVA 16. Tekstikenttä merkinnän lisäyksessä

Käyttäjä voi tarkastella aiemmin lisättyjä merkintöjä napauttamalla ”Edelliset merkinnät” yläpalkin valikosta. Merkinnät näytetään yksinkertaisessa listanäkymässä aikajärjestyksessä uusimmasta vanhimpaan ja käyttäjällä on myös mahdollisuus poistaa lisätty merkintä 24 tunnin sisällä merkinnän lisäämisestä painamalla roskakorikuvaketta. (Kuva 17.)



KUVA 17. Käyttäjän lisäämät edelliset merkinnät

7 JATKOKEHITYS

Mobiilisovelluksen kehitystä voidaan viedä erittäin pitkälle, mutta siihen vaaditaan yleensä enemmän kuin yhden henkilön kehitystyötä. Kehityksessä pitäisi ottaa huomioon sovelluksen testaaminen, jonka avulla saataisiin kartoitettua sovelluksen mahdolliset koodivirheet, virhetilanteet ja puutteet. Usein mobiilisovelluksien kehitystyössä on myös mukana UX-tiimi, joka suunnittelee käyttöliittymän ulkoasun ja navigoinnin. Yleisesti ottaen mobiilikehittäjät eivät ole kovin luovia käyttöliittymien suunnittelussa, joten olisi suotavaa, että UX-tiimi olisi sovelluksen jatkokehityksessä olla mukana.

Mobiilisovelluksen lisäksi järjestelmän käyttöä varten täytyy olla selainpohjainen käyttöliittymä, jonka avulla tutkijat voisivat järjestää tutkimuksia ja tutkia kerättävää dataa. Selainpohjaisessa käyttöliittymässä tulisi myös olla toiminnot datan vientiin, jotta dataa olisi helppo tutkia esimerkiksi Excelissä. Tutkijat voisivat luoda uusia tutkimuksia helposti selaimen käyttöliittymän avulla ja määritellä tutkimuskohtaiset kysymykset sekä päivän aikana ilmestyvät kysymykset.

7.1 Käyttäjätunnukset

Tutkittaville voisi olla myös selainpohjainen ympäristö, johon he pääsisivät kirjautumaan tunnuksilla ja katselemaan aikaisempia vastauksia sekä muokkaamaan tarvittaessa omia tietoja. Käyttäjät voisivat luoda halutessaan tunnukset palveluun suoraan mobiilisovelluksessa. Mobiilisovelluksen ensimmäisessä käynnistyksessä voisi olla vaihtoehtona joko kirjautua sisään, rekisteröityä tai osallistua tutkimukseen rekisteröitymättä. Rekisteröintiä varten käyttäjän tarvitsisi syöttää vain sähköpostiosoite ja salasana. Tunnusten avulla tutkittavalla voisi olla hallinta omista tiedoista ja hän voisi milloin tahansa kirjautua selaimen kautta hallintajärjestelmään katselemaan omia tietoja ja halutessaan myös poistaa tietonsa järjestelmästä.

7.2 Datan tallennus tietokantaan

Yksi ympäristön tärkeimpiä toimintaedellytyksiä on sille suunniteltu tietokanta, jota käytettäisiin datan hallintaan. Tässä projektissa oli perustoiminnot sisältävä

tietokanta käytössä, jonka avulla siirrettiin dataa JSON-muodossa. Tietokantaan tulisi tallentaa käyttäjien ID:t, tarvittavat tiedot, tunnukset sekä vastaukset kysymyksiin. Puhelimeen olisi hyvä saada myös reaaliaikaiset notifikaatiot, jotka pysyttäisiin lähettämään milloin tahansa tietokannasta suoraan esimerkiksi tietylle käyttäjäryhmälle.

7.3 Mobiilisovellus iOS-käyttöjärjestelmälle

Kaikilla tutkimukseen osallistuvilla ei luultavasti ole Android-käyttöjärjestelmällä varustettua puhelinta, sillä iPhone on Androidin lisäksi hyvin suosittu puhelin. Tätä varten pitäisi rakentaa iOS-ympäristössä toimiva sovellus, joka varmistaisi tutkimukseen osallistumisen myös iPhonella.

7.4 Järjestelmän käyttö testauksissa

Tutkimuspäiväkirjan käyttötapauksissa täytyy ottaa huomioon sen käyttö esimerkiksi erilaisten tuotteiden ja laitteiden pilotoinneissa ja testauksessa. Polar voisi testata helposti esimerkiksi jotain uutta tuotetta tai vaikka nykyisten tuotteiden uusia ominaisuuksia pienellä ryhmällä, hyödyntäen tutkimuspäiväkirja-sovellusympäristöä. Tutkimukseen osallistuvat käyttäjät voisivat esimerkiksi raportoida helposti mobiilisovelluksen avulla tuotteen tai ominaisuuden käytöstä, ongelmista ja erilaisista virhetilanteista. Tätä varten sovellukseen voisi kehittää lisätoimintoja, kuten kuvien liittäminen päiväkirjamerkintöihin.

7.5 Järjestelmän käyttöönotto

Koko järjestelmä tulisi ottaa vaiheittain käyttöön ja varmistaa mobiilisovelluksen toiminta eri laitteilla ja puhelinmalleilla ennen tutkimuksien järjestämistä. Mobiilisovellukseen voidaan lisätä hyvin paljon erilaisia, mutta hyödyllisiä ominaisuuksia, kuten esimerkiksi tutkittavien äänipalautte, kuvien lähettäminen ja sijaintitietojen tallennus kysymykseen vastatessa. Järjestelmään voisi mahdollisesti tulevaisuudessa liittää myös Polarin urheilukellot osaksi kokonaisuutta. Tutkittavan rannelaitteelta voitaisiin hakea esimerkiksi kysymyksen vastaushetkellä syketietoja, aktiivisuustietoja, unidataa sekä treenidataa.

8 YHTEENVETO

Opinnäytetyön tavoitteena oli suunnitella tutkimuspäiväkirjan arkkitehtuuri ja selostaa tutkimuspäiväkirjan käyttö todellisuudessa. Lisäksi tavoitteena oli toteuttaa alustava mobiilisovellus tutkimuspäiväkirjasta. Tuloksena syntyi kattava dokumentti, joka suunnittelee ja kuvaa koko järjestelmän toimintaa ja pohtii eri näkökulmia arkkitehtuurin ja toteutuksen kannalta. Mobiilisovelluksen ensimmäisellä versiolla pystyy osallistumaan tutkimukseen osallistumiskoodilla sekä täyttämään alkutiedot ja alkukysymykset. Näiden jälkeen mobiilisovellukseen tulee aina ilmoitus, kun käyttäjän täytyy vastata kysymyksiin. Sovelluksen päiväkirjaan voidaan myös kirjata merkintöjä, jotka siten lähetetään tietokantaan.

Järjestelmää ja arkkitehtuuria suunniteltiin haastattelemalla Polarin tutkimustiimin jäseniä sekä pohtimalla, miten järjestelmästä saataisiin mahdollisimman yksinkertainen ja helppokäyttöinen, mutta ominaisuuksien puolesta kuitenkin kattava. Raporttia ja mobiilisovellusta varten selvitettiin tutkimuspäiväkirjan käyttötarkoitus, olemassa olevat tutkimukseen soveltuvat sovellukset, erilaisten tutkimusten vaiheet sekä tutkimuksien avulla kerättävien tietojen käyttötarkoitukset.

Vielä ei ole päätetty tutkimuspäiväkirjan jatkokehityksestä, mutta saatujen palautteiden perusteella, tällaiselle järjestelmälle olisi kova tarve Polarilla. Tulevaisuudessa voisi olla mahdollista, että kehityksen jatkamista varten perustettaisiin pieni tiimi, joka ottaisi koko järjestelmän kehityksen ja pilotoinnin työn alle.

Opinnäytetyön lopputulos vastaa työalussa Polarin kanssa yhteistyössä määritellyjä tavoitteita. Opinnäytetyö opetti, kuinka pienistäkin ideoista saadaan isoja ja toimivia kokonaisuuksia aikaiseksi ja kuinka haastattelemalla eri henkilöitä, saa hyviä ideoita järjestelmän kehitykseen. Mobiilisovelluksen kehitys kehitti mobiilihojelmointitaitojani sekä opetti, kuinka oikeaa dataa saadaan helposti siirrettyä palvelimen ja mobiilisovelluksen välillä.

LÄHTEET

1. Keitä olemme. Polar Electro Oy. Saatavissa: https://www.polar.com/fi/tie-toa_polarista/keita_olemme. Hakupäivä 28.10.2020.
2. Dennis, Megan 2018. Native vs. Cross-Platform Apps – You’ll Be the Winner. Zeolearn. Saatavissa: <https://www.zeolearn.com/magazine/native-vs-cross-platform-apps-youll-be-the-winner>. Hakupäivä 28.10.2020.
3. Heller, Martin 2020. What is Kotlin? The Java alternative explained. Saatavissa: <https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html>. Hakupäivä 28.10.2020.
4. Guide to app architecture. 2020. Android Developers. Saatavissa: <https://developer.android.com/jetpack/guide>. Hakupäivä 28.10.2020.
5. Chugh, Anupam. Android MVVM Design Pattern. JournalDev. Saatavissa: <https://www.journaldev.com/20292/android-mvvm-design-pattern>. Hakupäivä 28.10.2020.
6. Favell, Andy 2020. The History of Git: The Road to Domination in Software Version Control. Welcome to the Jungle. Saatavissa: <https://www.welcometothejungle.com/en/articles/btc-history-git>. Hakupäivä 28.10.2020.
7. Android Studio. 2020. Wikipedia. Saatavissa: https://en.wikipedia.org/wiki/Android_Studio. Hakupäivä 28.10.2020.
8. Consuming APIs with Retrofit. Codepath. Saatavissa: <https://guides.codepath.com/android/consuming-apis-with-retrofit>. Hakupäivä 28.10.2020.
9. JSON: What It Is, How It Works, & How to Use It. 2020. Copter Labs. Saatavissa: <https://www.copterlabs.com/json-what-it-is-how-it-works-how-to-use-it/>. Hakupäivä 28.10.2020.