



Mobiilisovelluksen kehittäminen React Nativen avulla

Riikka Niemi

2021 Laurea



Laurea-ammattikorkeakoulu

Mobiilisovelluksen kehittäminen React Nativen avulla

Riikka Niemi

Tietojenkäsittely amk

Opinnäytetyö

05, 2021

Tietojenkäsittely

Tietojenkäsittely tradenomitutkinto (AMK)

Riikka Niemi

Mobiilisovellutuksen kehittäminen React Nativen avullaVuosi 2021 Sivumäärä 39

Mobiililaitteiden yleistymisen myötä älypuhelin on käytössä 96 prosentilla Suomen kotitalouksista. Mobiilisovelluksen hyödyntäminen on luonteva tapa käyttää palveluita ja tarjoaa toimijalle nykyaikaisen tavan lähestyä uusia käyttäjäryhmiä.

Tässä toiminnallisessa opinnäytetyössä kehitettiin prototyyppi liikunnallisesta mobiilisovelluksesta React Nativen avulla. React Native -kehitysalusta mahdollistaa sovelluksen kehittämisen eri mobiilikäyttöjärjestelmille samanaikaisesti.

Opinnäytetyön toimeksiantajana on Tuusulan kunnan liikunta- ja vapaa-ajanpalvelut. Prototyypin avulla toimeksiantajalle esiteltiin uusi tapa aktivoida kuntalaisia liikunnan pariin. Sovelluksen avulla voi osallistua ja ehdottaa aktiviteetteja, kuten juoksu- tai pyörälenkkejä muille käyttäjille.

Kehittämistyö on kuvailtu ohjelmointiprojektin vaiheiden kautta. Käyttäjätestin avulla todennettiin sovelluksen toimivuus ja kartoitettiin kuntalaisten kiinnostusta palvelun käytölle. Prototyyppi toimii pohjana sovelluksen jatkokehittämiselle.

Asiasanat: React Native, ohjelmointiprojekti, mobiilisovellus

Riikka Niemi

Mobile Application Development using React Native

Year	2021	Pages	39
------	------	-------	----

The share of smartphone owners in Finland reached 96 percent in 2020. This enables possibility to offer new services to wide target groups.

The aim of this Bachelor's thesis was to develop a functional prototype of a mobile application. The thesis was made in co-operation with Tuusula municipality Sports and Exercise services. As the result a new way to activate inhabitants to sports was presented to the mandator. With this application the user can browse and suggest exercises like biking or running for other users.

The thesis describes the development process of the mobile application with React Native platform. React Native development allows developing to multiple user interphases simultaneously.

The result were evaluated by mandator and user testing. The main purpose of the user testing was to evaluate mobile application usefulness for mandator and test product functionality in practice.

Keywords: React Native, development project, mobile application

Sisällys

1	Johdanto.....	8
2	Toiminnallisen työn lähtökohdat	8
2.1	Toimeksiantajan esittely	8
2.2	Opinnäytetyön rakenne ja tutkimusongelmat	9
2.3	Aiheen rajaus ja riskien arviointi	9
2.4	Keskeiset käsitteet.....	10
3	Ohjelmointiprojektin vaiheet.....	10
3.1	Tarpeiden kartoitus.....	11
3.2	Vaatusmäärittely.....	11
3.3	Suunnitteluvaihe	12
3.4	Toteutusvaihe	12
3.5	Testausvaihe	12
3.5.1	Käyttäjätestin suunnittelu ja testiympäristön määrittely	13
3.5.2	Käyttäjätestauksen suorittaminen	13
3.5.3	Testitulosten analysointi	13
3.6	Käyttöönotto.....	14
3.7	Projektin päättäminen	14
4	Teknologiat ja menetelmät.....	14
4.1	Hybridi vs. natiivikehittäminen	14
4.2	React Native	14
4.3	Expo	15
4.4	Tietokannat	15
4.5	Figma	15
5	Projektin toteuttamismalli.....	16
5.1	Aikataulu	16
5.2	Vaatusmäärittely.....	17
5.3	Käyttötapauskaavio	18
6	Prototyypin suunnittelu	18
6.1	Tunnistautuminen - uusi käyttäjä.....	19
6.2	Treenisivut	19
6.3	Infosivu	20
7	Työn toteutus	20

7.1	React Native ja Expo	20
7.2	React Native-tiedostorakenne.....	21
7.3	Natiivikomponentit	21
7.4	Sovelluksen ulkoasun muotoilu.....	22
7.5	Navigaatio.....	22
7.6	Formik	24
7.7	IOS- & Android-käyttöjärjestelmien eroavaisuuksien huomioiminen	25
7.8	Tietojen tallentaminen ja hakeminen tietokannasta	26
7.9	Prototyypin julkaisu Expon avulla	26
8	Mobiilisovelluksen testaus.....	27
8.1	Testiryhmä	28
8.2	Testikysymykset	28
8.3	Tulokset	29
8.4	Esiin tulleet haasteet.....	30
8.5	Testitulosten arviointi	31
8.6	Arviointi toimeksiantajan palautteen perusteella.....	32
9	Sovelluksen jatkokehittäminen.....	32
10	Pohdinta	33
	Lähteet.....	34
	Kuviot	36
	Liitteet	37

1 Johdanto

Tämän toiminnallisen opinnäytetyössä kehitetään React Nativen avulla prototyyppi liikunnallisesta mobiilisovelluksesta.

Toimeksiantajana toimii Tuusulan kunnan liikunta- ja vapaa-ajan palvelut. Prototyypin avulla toimeksiantajalle esitellään uusi palvelumuoto kuntalaisten aktivoimiseksi liikunnan pariin. Mobiililaitteiden yleistymisen kautta palvelu on kattavasti kuntalaisten käytössä ja tarjoaa mahdollisuuden tavoittaa uusia käyttäjäryhmiä.

Teoriaosuudessa selvennetään lukijalle ohjelmointiprojektin vaiheet ja kehitystyön keskeiset käsitteet. Kehitystyö esitellään ohjelmointiprojektin vaiheiden kautta. Prototyypille tehdään myös käyttäjätutkimus, jonka avulla selvitetään kuntalaisten kiinnostusta tämänkaltaisen palvelun käytölle sekä prototyypin toimivuus.

Lopuksi arvioidaan työn onnistumista testitulosten ja toimeksiantajan palautteen perusteella. Jatkokehityksaiheissa käsitellään ohjelmointiprojektin kehitysvaiheita prototyypistä valmiiksi tuotteeksi.

2 Toiminnallisen työn lähtökohdat

Opinnäytetyön ajatuksen taustalla oli tekijän oma liikuntaharrastus, jonka kautta oivallus uudenaikaisesta palvelusta sai alkunsa. Idea esiteltiin oman asuinpaikkakunnan Tuusulan vapaa-ajanpalveluille mahdollisuutena tarjota uutta palvelua kuntalaisten käyttöön. Toimeksiantajan mukaan tarve tämänkaltaiselle palvelulle oli tiedostettu ja vastaavanlaista palvelua oli suunniteltu vapaa-ajanpalveluiden käyttöön. Palvelu mahdollistaisi uusien käyttäjäryhmien tuomisen palveluiden pariin. Kehitystyö ideasta toimivaksi prototyypiksi toteutettiin yhteistyössä toimeksiantajan kanssa.

Teknologiaksi valikoitui React Native -kehitysalusta. React Native on suosituin hybridikehittämisen teknologia, jonka avulla voidaan kehittää puhelinsovellus samanaikaisesti useille käyttöliittymille. React Native kehittäminen perustuu React -komponenttikirjastoon ja tarjoaa mahdollisuuden tämän aiemmin opitun teknologian hyödyntämiseen ja sen laaja-alaisempaan hallintaan.

Mobiilikehittämisen ymmärrys ohjelmistokehittäjän työssä antaa mahdollisuuden hallinnoida sovelluksen koko ohjelmointiprojektia ja osoittaa osaamista prosessin hallinnasta. Kehitys yhdistää suunnittelun, taustalla tapahtuvien tiedonhallintajärjestelmien osaamisen sekä käyttäjälle näkyvän puolen hallinnan. Kannettavien laitteiden käyttö tulee edelleen lisääntymään tulevaisuudessa ja alan kehityksessä mukana pysyminen mahdollistaa monipuolisen työnkuvan. Mobiilikehittäminen osoittaa laaja-alaista osaamista työmarkkinoilla ja toimii erinomaisena työnäytteenä tulevaisuudessa.

2.1 Toimeksiantajan esittely

Tuusulan kunnan liikuntatoimi valikoitui toimeksiantajaksi tekijän asuinpaikkakunnan mukaan. Tuusulan kunnan liikuntatoimessa on panostettu kuntalaisten liikuntapaikkojen kehittämiseen ja ohjattuun liikuntaan. Lisäksi kuntalaisille tarjotaan tietoa liikunnasta ja erilaisia kotitreenejä. Mobiilipalveluita ei liikuntapalveluissa olla vielä hyödynnetty.

Tuusulan kunnan vapaa-ajan palveluiden tavoitteena on tarjota kuntalaisille monipuolisia saavutettavia liikuntamahdollisuuksia. Tavoitteena on tarjota tukea ja motivoida kuntalaisia löytämään itselleen sopivia liikuntamuotoja ja sitä kautta ylläpitämään työ- ja toimintakykyä. Kuntalaisille halutaan tarjota itselle sopivia liikuntamuotoja ja sitä kautta valmiudet liikkua omatoimisesti. (Tuusula/Liikunta 2021).

Tuusulan kunnan liikuntatoimessa oli jo ennen idean esittämistä suunniteltu mobiilipalvelun tarjoamista kuntalaisten käyttöön. Tarve uudenlaisen palvelun kehittämiseksi oli siis jo tiedostettu. Mobiilipalvelun tavoitteena on palvella uusia kohderyhmiä ja etenkin nuorempia kuntalaisia. Opinnäytetyön tuotteena syntyvä mobiilisovellus vastaa tähän tarpeeseen. Prototyypin avulla on tarkoitus esitellä uusi palvelu toimeksiantajalle.

2.2 Opinnäytetyön rakenne ja tutkimusongelmat

Opinnäytetyössä kehitetään uusi tuote ideasta toimivaksi prototyypiksi. Lukijalle kuvataan ohjelmointituotantoprosessin kulku mobiilikehittämisessä ensin teoriassa ja myöhemmin käytännön toteutuksen kautta. Tuotoksen onnistumista arvioidaan käyttäjätestauksen ja toimeksiantajan palautteen perusteella.

Työ etenee ohjelmointikehityksen teoriaosuudesta keskeisimpien käsitteiden selventämiseen lukijalle. Kehitysprojektissa esitellään aluksi työlle asetetut vaatimusmäärittelyt. Tämän jälkeen esitellään työn suunnittelu kuvallisessa muodossa. Kehitysvaiheessa tarkastellaan erityisesti React Native -mobiilikehittämisen erityispiirteitä ja esille tulleita haasteita. Käyttäjätutkimuksen avulla testaajat pääsevät kokeilemaan prototyyppiä omilla mobiililaitteillaan ja palautteen avulla varmistetaan prototyypin toimivuus ja kiinnostus palvelun käyttöä kohtaan. Lopuksi esitetään näkökulmia prototyypin jatkokehittämiseen ja julkaisemiseen.

Opinnäytetyö vastaa kysymyksiin:

- Mitkä ovat tyyppillisen mobiilisovelluksen ohjelmointiprojektin vaiheet?
- Miten mobiilisovellus toteutetaan React Nativen avulla?
- Olisiko mobiilisovelluksen käytölle kiinnostusta Tuusulan kuntalaisten keskuudessa?

2.3 Aiheen rajaus ja riskien arviointi

Aihe rajataan React Native -mobiilisovelluksen ohjelmointiprojektin vaiheisiin vaatimusmäärittelyyn, suunnitteluun, prototyypin toteuttamiseen ja testaamiseen. Työssä toteutetaan toimiva prototyyppi liikunnallisesta mobiilisovelluksesta. Tavoitteena ei ole täysin valmis tuote, mutta työ toimii perustana jatkokehitykselle ja mobiilisovelluksen julkaisulle. Lopuksi prototyypin toiminta testataan pienimuotoisesti käyttäjillä ja saadun palautteen perusteella esitellään hankeidea toimeksiantajalle.

Projektin eri vaiheisiin liittyy erilaisia riskejä projektin toteutumisen kannalta. Suunnitteluvaiheessa aiheen rajaus on erityisen tärkeää, ettei projekti osoittaudu myöhemmin liian kunnianhimoiseksi. Aika ja tietoresurssien tulisi olla hallittavissa. Haasteen projektille tuo uusien teknologioiden omaksuminen. Toteutuksen kannalta teknologioiden oppiminen ja ohjelmointiin käytetty aika on suunniteltava etukäteen, eikä aikataulua kannata suunnitella liian tiukaksi. Työn testauksen haasteena on valitun Expo CLI julkaisualueen toimintavarmuus testin aikana.

Mobiilisovelluksen kehittäminen on haastava kokonaisuus aina toiminnallisuuksien suunnittelusta julkaisuun ja prosessi voi useammaltakin kehittäjältä viedä jopa puoli vuotta. Aihealueen rajauksessa onkin todella tärkeää, että työ pystytään toteuttamaan järkevällä aikataululla. Tästä syystä mobiilisovelluksen kehittäminen on vaiheistettu osa-alueisiin, joista tässä työssä kuvaillaan toteuttamisprojektin ensimmäinen vaihe.

Korona-ajan rajoitukset asettavat oman haasteensa projektille. Testaaminen ja koronan aiheuttamat kokoontumisrajoitukset sekä testaajien mahdollinen epävarmuus kokeilla sovellusta tartunnan pelossa, tulee ottaa huomioon riskejä arvioitaessa. Asiaa voi tarkastella myös positiiviselta kannalta. Rajoitusten voimassaolon aikana on hyvä aika rauhassa kehittää ja suunnitella mobiilisovellus, joka voidaan saattaa loppuun ja lanseerata rajoitusten päättyttyä.

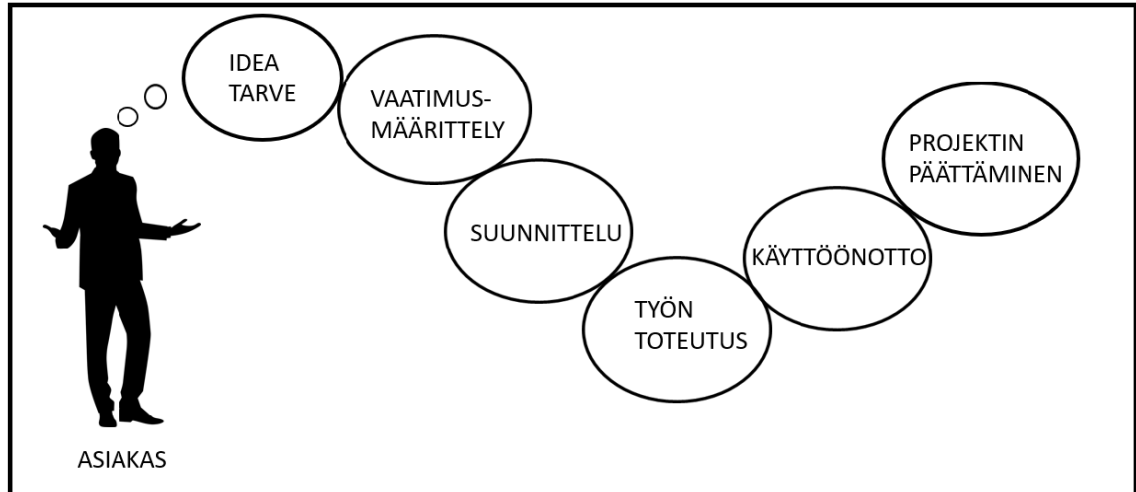
2.4 Keskeiset käsitteet

JavaScript	Ohjelmointikieli, joka mahdollistaa dynaamisen, interaktiivisen sisällön lisäämisen verkkosivuille, esim. lomakkeet.
Käytettävyys	Käyttäjän arvio ohjelmiston toimivuudesta ja helppokäyttöisyydestä sitä käytettäessä.
Mobiilisovellus	Mobiililaitteeseen ladattava ohjelmisto tai ohjelma. Mobiililaitteita ovat matkapuhelimet, älykellot ja tabletit.
Ohjelmointiprojekti	Tehtävä kokonaisuus, joka koostuu eri vaiheista ja sillä on selkeä alkamis- ja päättymisajankohta. Sisältää vaiheet: Aloitust, suunnittelu, toteutus ja päättäminen.
Ohjelmointiprosessi	Ei selkeää päättymisvaihetta. Ohjelmistoprojektin kehitys- ja hallintatapa.
Projektimalli	Tapa, jolla työvaiheet jaksotetaan. Esimerkiksi vesiputous- ja ketterän kehityksen -mallit.
Prototyyppi	Rami Juvosen (2018) mukaan asiakas odottaa prototyypin olevan usein lähellä lopullista tuotetta. Prototyyppi voi olla kertakäyttöinen prototyyppi esittelyä varten tai evoluutiotyyppi, jota on tarkoitus jatkokehittää.
React	JavaScript komponenttikirjasto käyttöliittymien toteuttamiseen verkkoselaimessa. Mahdollistaa uudelleen käytettävien komponenttien käytön ja latautuu nopeammin virtuaalisen ohjelmointirajapintansa ansiosta päivittämällä ainoastaan muuttuneet komponentit koko käyttöliittymän sijaan.
Wireframe	Rautalankamalli, visuaalinen tapa mallintaa ja suunnitella tuotteen toimintoja ja ulkoasua.

3 Ohjelmointiprojektin vaiheet

Projekti käynnistetään yleensä tarpeen pohjalta ja sille on tunnuksenomaista selkeä tavoite. Erilaisia projekteja ovat esimerkiksi tuotekehitys-, tutkimus- ja kehittämisprojektit. Projektin kustannuksia tai siihen kohdistuvaa työmäärää voidaan tarkastella sen alkamisajan ja loppumisen väliin jäävän elinkaaren vaiheiden kautta. Elinkaariajattelumallin vaiheet kertovat mitä ohjelmointiprojektissa tehdään, ei miten toteutus etenee. (Mäntyneva 2016.)

Ohjelmistoprojektin projektimallit poikkeavat vain yksityiskohdiltaan tyypillisestä ohjelmistoprojektin rakenteesta. Ohjelmointiprojektin projektimalli etenee tarpeiden kartoituksesta, työn määrittelyyn, suunnitteluvaiheeseen, työn toteutukseen, käyttöönottoon ja projektin päättämiseen. (Juvonen 2018, 13.) Kuviossa 1 olen havainnollistanut tyypilliset ohjelmointiprojektin vaiheet.



Kuvio 1: Ohjelmistoprojektin vaiheet

3.1 Tarpeiden kartoitus

Ohjelmointiprojekti alkaa usein asiakkaan tarpeiden kartoittamisella. Tarpeiden kartoitus vastaa kysymykseen: Miksi järjestelmä pitää tehdä? Asiakkaan tarvekartoitusta voidaan kutsua myös esitutkimusvaiheeksi. Tarvekartoituksen suurin haaste on asiakkaan todellisten tarpeiden ymmärtäminen ja priorisointi. (Haikala & Mäkijärvi 2006, 37.)

Tarvekartoituksessa on tärkeää lähtötilan ja tavoitetilan ymmärtäminen kohderyhmän kannalta. Kohderyhmä koostuu tuloksia hyödyntävistä ihmisryhmistä ja organisaatioista. Tuotteen käyttäjien tarpeet, ongelmat, halut, toiveet ja vaatimukset määrittelevät tuotteen ominaisuudet. Toteutettavat ominaisuudet määritellään arviointikriteerien kautta. (Santanen 2017.)

3.2 Vaatimusmäärittely

Määrittelyvaihe vastaa kysymykseen: ”Mitä järjestelmä tekee?” Asiakasvaatimuksien analysointi toimii pohjana ohjelmistovaatimuksien määrittelylle. Ohjelmistovaatimukset pitävät sisällään toteutettavan järjestelmän järjestelmävaatimukset ja sen ominaisuudet. (Haikala & Mäkijärvi 2006, 38.)

Toiminnallinen vaatimusmäärittely sisältää tuotteen käyttäjälle mahdollisten toimintojen määrittelyn. Se vastaa kysymykseen: ”Mitä sovelluksella on mahdollista tehdä?” Toiminnalliset vaatimusmäärittelyt esitellään asiakkaalle usein käyttäjätarinan tai kaavion avulla. Toiminnallisen vaatimusmäärittelyn tulee myös sisältää tieto asioista, joita toimintoihin ei sisällytetä. Tämä auttaa asiakasodotusten hallinnassa ja tuo yhteisymmärryksen valmiin tuotteen vaatimuksista. (Dooley 2011, 42-43.)

Tekninen määrittely sisältää tuotteen tekniset ominaisuudet ja käytettävät teknologiat. Teknisissä vaatimuksissa määritellään esimerkiksi ohjelman nopeusvaatimus, tietojen tallentaminen ja turvallisuuteen liittyvät vaatimukset. Ohjelmistolle asetetut vaatimukset tulee voida testata. Vaatimusmäärittelyyn liittyy olennaisesti vaatimusten tarpeellisuuden analysointi ja priorisointi. (Dooley 2011, 44-46.) Priorisointia voidaan hyödyntää aikataulun tai budjetin uhatessa ylittyä toteuttamaan silti toimiva tuote. 80:20-säännön mukaan 20% tekemisestä tuottaa 80% tuloksista ja 80% käyttäjistä tulee toimeen 20% ominaisuuksista. (Juvonen 2018, 91-92.)

3.3 Suunnitteluvaihe

Suunnittelun tavoitteena on muuntaa asiakkaan tarpeen mukaan määritelty tuote järjestelmän toteutuksen kuvaukseksi (Haikala & Mäkijärvi 2006, 81.)

Luonnostelu on yleisesti käytössä oleva tekniikka idean alkuvaiheen konkretisoimiseksi. Luonnostelun tavoitteena on esitellä yksinkertaistetulla tavalla tuotoksen idea ja toiminta. Yksityiskohtaisempaan ja realistisempaan suunnitteluun siirryttäessä ulkoasun suunnittelussa käytetään usein Wireframe- eli rautalankamallia. Kehitystyötä varten tulee suunnitella selkeä suunnitelma käyttäjän liikkumisesta näkymästä toiseen (Perea & Giner 2017, luku 3, luku 5.) Figma on helppokäyttöinen suunnittelutyöväline rautalankamallin teosta aina toiminnallisten mallien suunnitteluun ja mallinnukseen (Bracey 2018).

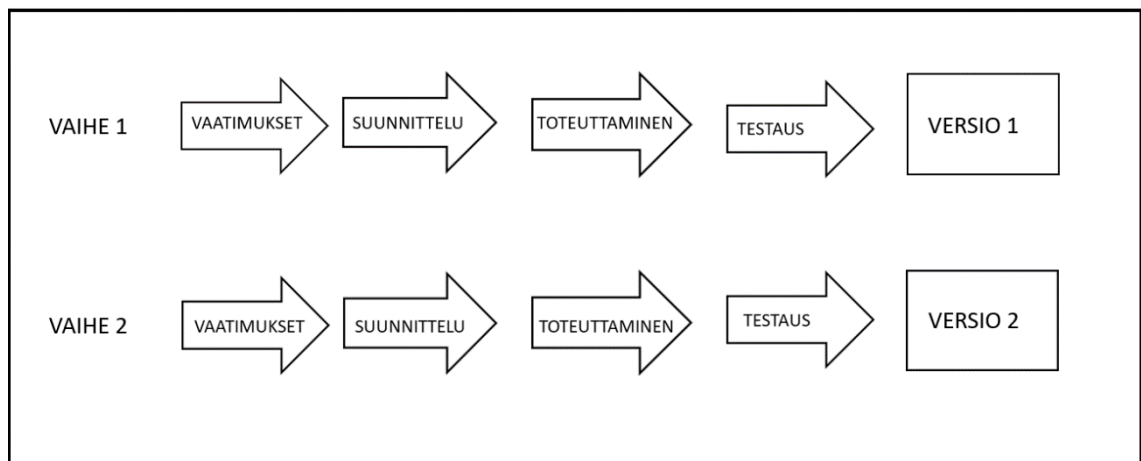
3.4 Toteutusvaihe

Ohjelmistoprosessi kertoo miten ohjelmistoprojektia kehitetään ja hallitaan. Projektimalli on malli, jonka mukaan ohjelmointiprojekti etenee. Toteutusvaiheen menetelminä ohjelmointikehittämisessä voidaan käyttää apuna erilaisia projektimalleja. (Münch & Armbrust 2012, luku 1)

Varhaisimpia ja yksinkertaisimpia malleja edustaa 1970-luvulla kehitetty lineaarisesti etenevä vesiputousmalli. Vesiputousmallissa työn vaiheet suunnitellaan tarkasti jo projektin alkuvaiheessa. Työ suoritetaan peräkkäisinä vaiheina, eikä aikaisempiin vaiheisiin ole tarkoitus palata. (Münch, & Armbrust 2012, luku 1)

Tarpeeseen joustaa alkuperäisistä suunnitelmista vastasivat ketterän kehityksen projektimallit 2000-luvun alussa. Parhaiten tunnettu ketterän kehittämisen malli on Scrum. Malli etenee työlistan pohjalta laadittujen työkokonaisuuksien suorittamisella noin 1 - 4 viikon sprintteinä. Toteutus etenee yksi ominaisuus kerrallaan, joista muodostuu joustava kokonaisuus. (Juvonen 2018, 15-25.)

Evoluutiivinen prototyyppi -kehittäminen on suosittu tapa kehittää opiskeluprojekteja. Hyötynä on saavuttaa nopeasti näkyvä testattava tuote valmiiksi ja saadun palautteen perusteella kehittää tuotetta edelleen. Kehittämisen vaihe muistuttaa vesiputousmallia, mutta seuraavan vaiheen suunnittelu muokataan saadun palautteen perusteella ketterän mallin mukaisesti. Erona ketterään kehittämiseen, mallissa on tavoitteena yksittäisten toiminnallisuuksien lisäämisen sijaan kehittää testattava kokonaisuus. (Dawson 2009, 126-127.) Kuviossa 2 on kuvattu Evoluutiivisen kehittämisen malli.



Kuvio 2: Evoluutiivisen kehittämisen malli

3.5 Testausvaihe

Testauksen työvaiheita ovat testin suunnittelu, testiympäristön luominen ja testin suorittaminen sekä testitulosten analysointi. Mikäli testaja toimii ohjelman tekijä, on tavoitteena

osoittaa ohjelman toimivuus. Ohjelmistojen testaus on kuitenkin suunnitelmallista virheiden etsintää. Käytettävyydestestauksen avulla varmistetaan, että käyttäjä kykenee selviämään tehtävistä, joita varten järjestelmää ollaan rakentamassa. Testaajien tulee olla kehittämistyöstä riippumattomia testaajia ja tuotteen mahdollisia loppukäyttäjiä. (Haikala & Mäkijärvi 2006, 284, 290-291.)

Puhelinsovellusta voidaan testata erilaisten mobiilisimulaatioiden näkymien kautta tietokoneen avulla. Paras tapa testata toimintoja loppukäyttäjien avulla tapahtuu kuitenkin fyysisellä puhelinlaitteella. (McWherter & Gowell 2012, 113-115.) Mobiilisovellusten testauksessa voidaan testausta tehdä ns. lasilaatikko (white box) -testauksena, jolloin testaajalla on tietoa ohjelman toteutuksesta. Tällöin testauksessa keskitytään kooditason testaukseen. Mustalaatikkotestauksessa testaajalla ei ole lainkaan ennakkotietoa ohjelmiston toiminasta tai rakenteesta. Mustalaatikkotestauksessa testaaja käyttää yleensä käyttöliittymää testaustehtävien suorittamiseen. (Hamza & Hammad 2020.) Expo Go -sovelluksen avulla voidaan kehitettyä sovellusta käyttää testaajan omalla laitteella. Sovelluksen avulla voidaan todentaa kehitetyn tuotteen toimintoja ja interaktiivisuutta. (Expo Dokumentaatio 2021).

3.5.1 Käyttäjätestin suunnittelu ja testiympäristön määrittely

Testin suunnittelussa määritellään mitä, missä ja miten tuotetta testataan. Kehittämistyön loppupuolella voidaan testin avulla varmistaa, että tuote vastaa vaatimuksia. Alkuvaiheessa testausta voidaan käyttää yksittäisten toimintojen testaamiseen tai kehittämisen suunnan määrittelyyn. Testauspaikka riippuu testin tavoitteesta ja testattavasta tuotteesta. Paikka voi vaihdella laboratorio-olosuhteista aina etätestaukseen. Testauksessa voidaan myös yhdistellä kontrolloidussa ympäristössä ja testaajan omassa ympäristössä suoritettuja testejä.

Tavoitteiden perusteella määritellään miten testi tehdään. Kehitettyä tuotetta voidaan testata testaajalle annettujen tehtävien avulla. Omaa tuotetta voidaan myös vertailla kilpailijan tuotteeseen. (Barum 2011, 111-113.) Tilastojen mukaan noin 83 prosenttia virheistä ohjelmistossa voidaan havaita jo 5 testaajan avulla. Mobiilitestauksessa tulee huomioida ohjelmiston toiminnan testaus eri käyttöjärjestelmillä. (Myers, Sandler & Badgett 2012, 149, 219).

Testisuunnitelmalle laaditaan kirjallinen suunnitelma. Suunnitelmassa tulee ilmi testiryhmän rekrytointisuunnitelma ja vaatimukset. Lisäksi määritellään tutkimuksen tavoite, tiedonkeruumenetelmät, mittausmenetelmät ja aikataulu. Testin valmistelu pitää sisällään myös kyselylomakkeen tai tarkkailumetodien laatimisen ja osallistujien rekrytoinnin. (Barum 2011, 147, 157-158.)

3.5.2 Käyttäjätestauksen suorittaminen

Ennen varsinaista testiä voidaan tehdä alustava pilottitestaus, jonka avulla varmistetaan, ettei testin aikana tulisi eteen odottamattomia yllätyksiä. Varmistetaan esimerkiksi, että kyselylomake on yksiselitteinen ja mittaa haluttuja asioita. (Barum 2011, 191.)

Testi suoritetaan suunnitelman mukaisesti. Testiryhmälle selostetaan testin eteneminen ja ryhmä ohjeistetaan testin tekemiseen. Valvotussa testissä monitorioijan toimintaan tulee kiinnittää huomiota, että testitilanne olisi kaikille testaajille samanlainen. Testaajia neuvotaan suunnitelman mukaisesti. Testi voidaan päättää kyselylomakkeen täyttämiseen tai haastattelun. (Barum 2011, 231-233.)

3.5.3 Testitulosten analysointi

Testituloksien jäsentely voidaan tehdä ylhäältä-alas, järjestämällä tulokset ennalta määriteltyihin kategorioihin. Jäsentely alhaalta ylös tapahtuu liittämällä löydökset ryhmiin samankaltaisten tulosten perusteella. Tulosten esittäminen numeraalisesti on suositeltavaa etenkin pienemmillä tutkimusjoukoilla prosentuaalisten tulosten sijaan. Tuloksen esittäminen esimerkiksi kolme vastaajaa viidestä antaa lukijalle realistisemmän kuvan tuloksesta, kuin 60% vastaajista.

Määrällinen tulosten esittäminen on numeraalisesti mitattavaa, kuten tehtävään kulunut aika tai virheiden määrä. Laadulliset havainnot kirjoitetaan monesti ylös ja yhdistellään eri kategoriaihin. Kyselylomake voi tuottaa sekä määrällisiä, että laadullisia tuloksia. Likertin viisiportaista asenne-asteikkoa arvioitaessa on hyvä huomioida, että ihmiset usein välttävät ääripään arvioita. Havainnot voidaan myös luokitella niiden vakavuuden (vakava virhe toiminnossa) tai paikallisuuden (virhe tapahtuu tietyssä toiminnossa useilla testaaajilla) mukaan. Käyttäjätutkimuksen tulosten perusteella voidaan esitellä suosituksia testattavan tuotteen korjaamiseksi tai kehittämistä varten. Tulokset voidaan esittää raportin, taulukon tai suositusten muodossa. (Barum 2011, 268-267, 278.)

3.6 Käyttöönotto

Käyttöönottovaiheen työmäärä monesti aliarvioidaan. Käyttöönottovaiheeseen liittyy henkilöstön koulutus, käyttäjätuki, käyttöohjeiden laadinta ja mahdollinen muutosvastarinnan voittaminen. Asiakkaan tuki on olennaista käyttöönottovaiheessa. Käyttöönottovaiheessa varmistetaan, että projektia tullaan käyttämään. Mikäli projekti epäonnistuu, tulee arvioida hyödynnettävien tuotosten käyttämisen mahdollisuus projektin ulkopuolella. Projektin käyttöönottovaiheessa tuotokset tulevat näkyviksi ja tämän takia ammattitaitoinen käyttöönotto on tärkeää. (Juvonen 2018, 107-108.)

3.7 Projektin päättäminen

Projekti päätetään asiakkaan kanssa varmistamalla, että projektisopimuksen mukaiset tuotokset on toimitettu. Asiakkaalle toimitetaan testatut viimeisimmät versiot lähdekoodista tai valmiiksi käännettyistä ohjelmistoista. Mikäli projektille ei ole luvassa jatkoa, projektia varten hankitut käyttöoikeudet ja tietojärjestelmiin liittyvät tilit tulee sulkea ja siirtää asiakkaalle. Poikkeuksena mahdollisen ylläpitotiimin oikeudet. (Juvonen 2018, 110.)

4 Teknologiat ja menetelmät

Johdannon luvussa 2.5 on käsiteluttelo, jossa esitellään lyhyesti tässä työssä käytettyjä käsitteitä lukijalle. Tässä luvussa avataan tarkemmin kehittämistyössä käytettyjä keskeisimpiä teknologioita ja menetelmiä. Kaikille käsitteille voi olla vaikea löytää suomenkielistä vastinetta tai suomenos voi olla harhaanjohtava. Tämän takia olen liittänyt joidenkin käsitteiden perään niiden englanninkielisen vastineen.

Tietotekniikan nopean kehityksen takia englanninkielisten lainasanojen käyttö on yleistä ja suomenkielisten termien käyttö vakiintuu hitaasti. Englanninkielisten lainasanojen käyttö on yleistä ja suora käännös saatetaan tulkita väärin. (Seppälä 2021.)

4.1 Hybridi vs. natiivikehittäminen

Mobiilisovelluskehitys voidaan jakaa web-, natiivi- ja hybridikehittämiseen. Web-kehittämisellä tarkoitetaan ensisijaisesti verkkosivuille suunniteltuja sovelluksia, joita voidaan käyttää myös mobiililaitteissa. Natiivi-kehittäminen on sovelluksen kehittämistä vain yhdelle käyttöjärjestelmälle sopivaksi. Natiivikehittämisen etuna on paras käytettävyyys, kun kehittämisessä voidaan hyödyntää juuri kyseessä olevan käyttöjärjestelmän teknologioita. Mobiilikäyttöjärjestelmiä ovat esim. Googlen Android ja Applen iOS sekä Windows. Hybridikehittäminen tapahtuu soveluskehitysalustan avulla, jolloin kehittäminen tapahtuu usealle käyttöjärjestelmälle samanaikaisesti. (Jabangwe, Edison & Duc. 2018.) Hybridisovellusalan avulla voidaan käyttää natiivitoimintoja esim. verkkonäkymien avulla tai tuomalla käyttöön käyttöjärjestelmän omia komponentteja (O'Reilly 2021).

4.2 React Native

Facebookin kehittämä React Native on ohjelmoinnin sovelluskehys (framework) mobiilisovellusten kehittämiseen JavaScript-ohjelmointikielillä, jossa käytetään React JavaScript kirjastoja (Dabit 2019, 26.) React Nativen avulla voidaan kehittää järjestelmäriippumattomia

mobiilisovellutuksia, jotka yltyvät vain yhtä järjestelmää varten kehitettyjen sovellusten tasolle. Sen käyttö nopeuttaa kehitystyötä, tekee siitä edullisempaa ja vähentää merkittävästi resurssien tarvetta. React Nativen suosiota lisää sovellusten käyttäjävällisyys ja panostus ulkoasun kehittämisessä. (Rodriguez Martinez 2018, 5; Boduch 2017, 210-212.) Statistan tutkimuksen mukaan React Native oli vuosien 2019 ja 2020 suosituin hybridisovellusten kehittämisalusta ja noin 42% ohjelmointikehittäjistä käytti sitä (Liu 2020).

4.3 Expo

Kehitystyössä hyödynnetään React-sovelluksille tarkoitettua Expo-sovellusalustaa ja -viitekehystä. Expo on React Nativelle kehitetty avoimen lähdekoodin sovellusalusta, jonka avulla voi kehittää natiiveja iOS ja Android sovelluksia JavaScript ohjelmointikielellä. Expo-cli komentokehotteiden avulla voi luoda tai julkaista uuden React Native-projektin. Expo kehitystyökalun avulla React Native-sovelluksen ulkoasua ja toimintoja voidaan tarkastella ja testata kehitystyön aikana omassa puhelimessa Expo Go-applikaation avulla ja tietokoneen puhelin-simulaattorin avulla. Expo mahdollistaa myös sovelluksen ilmaisen julkaisemisen omalla serverillään testausta varten. Expo helpottaa kehitystyötä, mutta asettaa myös rajoituksia sovelluksen koolle ja toiminnoille. (Expo Dokumentaatio 2021).

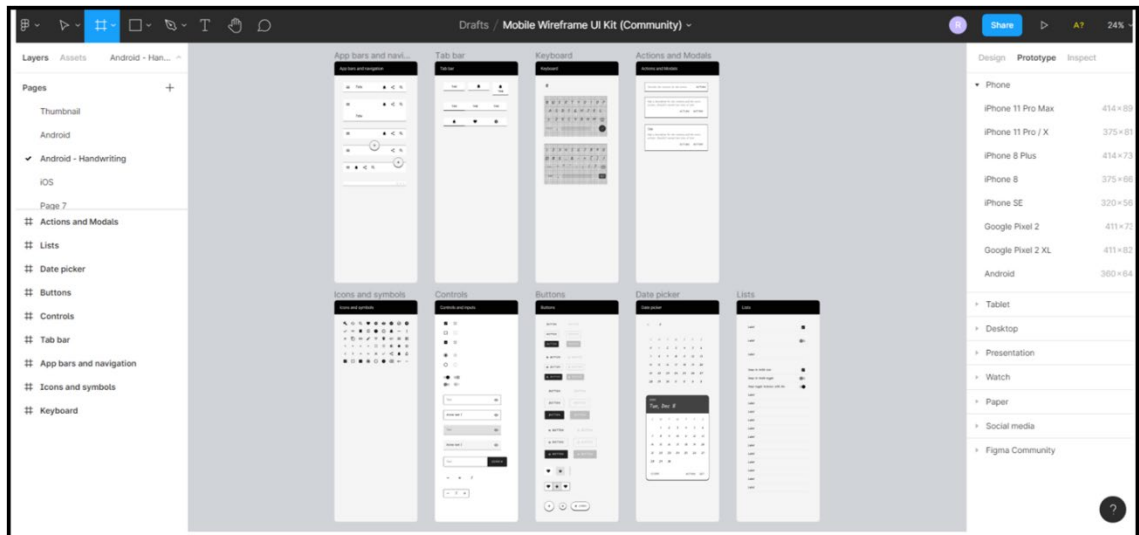
4.4 Tietokannat

Tietokannanhallintajärjestelmä eli Database Management System, on ohjelmisto, jonka avulla voidaan käsitellä, hallita, hakea ja tallentaa tietokannassa olevaa tietoa. Tietokannan tietomalli voi perustua olio-, relaatio-, hierarkia- tai tiedostomalliin. (Foster 2016, 3-8). Tässä työssä päädyttiin käyttämään MongoDB-tietokantaa, koska sen käyttö oli tuttu aiemmista opinnoista. MongoDB on myös vakiinnuttanut paikkansa osana mobiilikehittämistä.

MongoDB-tietokannanhallintaohjelmisto tallentaa tiedon NoSQL-rakenteen pohjalle ja tieto tallennetaan JSON-muodon kaltaisesti. MongoDB mahdollistaa monenlaisen tiedon tallentamisen ja yhdistämisen samaan tiedostoon, kuten kuvat, videot, teksti tai sijaititiedot. Myös tietokannan rakenteen voi määritellä melko vapaasti, etenkin verrattuna jäykempiin taulukkopohjaisiin relaatiotietokannan malleihin. (Botelho 2021). Mongoose on MongoDB:n (Object Document Mapping) kirjasto, joka selkeyttää tietokannan käyttöä ja yhteydenottoa. Mongoosen avulla voidaan määritellä tallennettavan tiedon rakenne ja oletusarvot. Lisäksi Mongoose tarjoaa rajapinnan tietokantaan, kun halutaan luoda, lukea, päivittää tai poistaa siellä olevaa tietoa. (Karnik 2018.)

4.5 Figma

Figma on mallinnukseen kehitetty tietokoneohjelma. Suunnitteluprojekti tehdään suoraan verkkoselaimessa. Verkossa kehittäminen mahdollistaa suunnittelutyön jakamisen, kommentoinnin ja muokkaamisen ryhmän jäsenten kesken sekä suunnittelumallin tarkastelun eri laitteissa. Suunnittelualusta tarjoaa valmiita mallipohjia ja erilaisia komponenttikirjastoja suunnittelijan käyttöön. Myös erilaisten tiedostojen kuten kuvien ja videoiden liittäminen malliin on mahdollista. (Hall 2015.) Suunnittelija voi luoda yhteyksiä eri näkymien välille ja havainnollistaa toimintoja ja käyttäjän kulkureittejä sovelluksessa. Figman avulla voidaan myös tuottaa esim. grafiikoita ja tyylitiedostoja varten valmista koodia. (Bracey 2021.) Kuviossa 3 näkyy Figma-suunnittelutyökalu ja erilaisia mobiilisovellusten suunnittelussa käytettyjä valmiita komponentteja sekä eri puhelinmallien mallipohjia.



Kuvio 3: Figma-suunnittelutyökalu

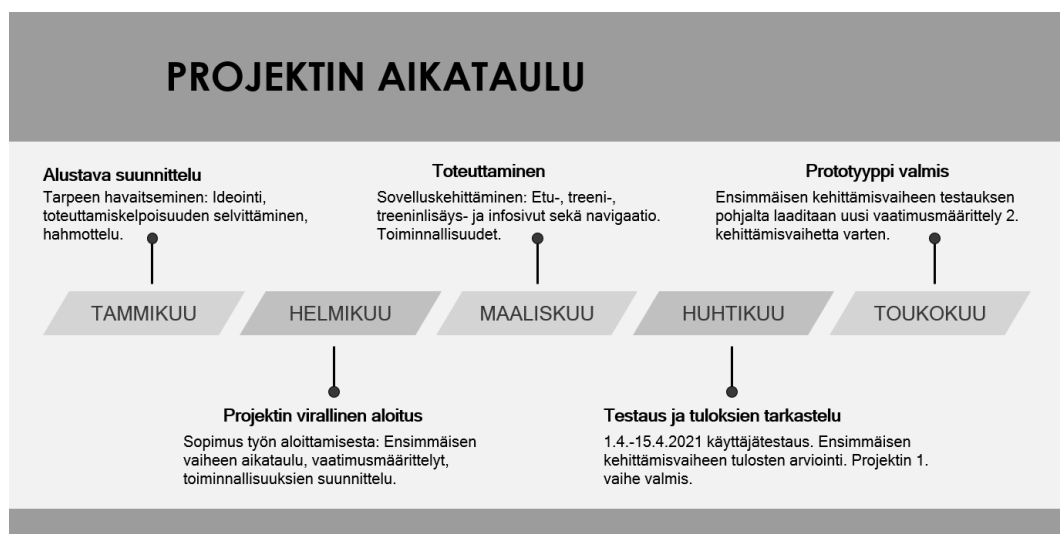
5 Projektin toteuttamismalli

Kehittämistyössä päätin noudattaa evolutiivisen kehittämisen projektimallia. Malli mahdollistaa kehittämisen vaiheittain ja antaa mahdollisuuden muokata suunnitelmaa saadun palautteen kautta. Evolutiivisen prosessimallin etuna on näkyvien tuloksien saavuttaminen lyhyelläkin aikataululla. Projektimalli on esitelty tarkemmin edellä luvussa 3.

Tässä työssä esitelty mallin ensimmäinen kehitysvaihe päättyy prototyypin testaamiseen ja tulosten analysointiin. Toisessa vaiheessa, jatkokehittämisen suunnittelua tarkennetaan testauksen tuloksena saadun palautteen perusteella. Sovelluksen julkaiseminen on mielestäni mielekästä erotella omaksi kolmanneksi kehittämisen vaiheekseen.

5.1 Aikataulu

Projektin toteuttamisen aikataulu oli haasteellista arvioida ilman aiempaa kokemusta kehittämistyöstä React Nativella. Myös mahdollisiin yllättäviin ongelmiin kehitystyön aikana on hyvä varautua. Tämän vuoksi ensimmäisen vaiheen toteutus suunnitelmaan on sisällytetty vain välttämättömimmät testauksen mahdollistavat toiminnot. Ensimmäisen vaiheen aikataulu on esitelty kuviossa 4.



Kuvio 4: Projektin aikataulu

5.2 Vaatimusmäärittely

Sovelluksen toiminnallisuudet esiteltiin ja hyväksyttiin toimeksiantajalla. Yhteistyötä tehtiin sähköpostien välityksellä. Esittelyn perusteella toimeksiantaja ehdotti sovelluksen ulkoasun toteuttamista Tuusulan kunnan graafisten ohjeiden mukaisesti. Muilta osin toimeksiantaja ei asettanut toiveita vaatimusmäärittelylle.

Vaatimusmäärittelyssä keskityin erityisesti prototyypin toiminnallisiin vaatimuksiin. Teknisissä vaatimuksissa määritellään lähinnä käytetyt teknologiat ja testaamisen mahdollistaminen testaajan omalla iOS- tai Android- puhelimella. Yleisissä vaatimusmäärittelyissä huomioidaan helppokäyttöisyys ja ulkoasun määrittely. Toiminnallisissa vaatimusmäärittelyissä näkyvät toiminnot, jotka pitää olla valmiina prototyypin testausvaiheessa. Käyttäjällä tulee olla mahdollisuus päästä etu-, trenni- ja infosivuille ja lisätä trenniehdotus. Testauksen palautteenantoa helpottamaan lisättiin kyselylomakkeen linkki sovelluksen infosivulle.

Toiminnallisia vaatimusmäärittelyjä tehtiin myös seuraavaa kehitysvaihetta varten yleisellä tasolla, mutta evolutiivisen prototyypimallin tavoitteena on tarkentaa toisen kehitysvaiheen vaatimukset testauksessa saadun palautteen perusteella. Kirjautuminen jätettiin seuraavaan kehitysvaiheeseen, koska käyttäjätestauksen tulee olla testaajalle mahdollisimman helppoa, eikä testaajien tietoja ole tarpeen tallentaa tietokantaan. Tämä vuoksi myös trenniin liittymisen ja poistamisen -toimintoja ei voitu myöskään liittää ensimmäiseen kehitysvaiheeseen.

Kirjautumista koskevissa vaatimusmäärittelyissä tulee huomioida turvallisuuteen ja lainsäädäntöön liittyvät vaatimukset. Sähköpostiosoitteen turvallinen lähettäminen ja tallentaminen tietokantaan sekä käyttäjän suostumus tietojen tallentamiseen tulee huomioida. Vaatimusmäärittely on esitelty kuviossa 5.

Vaatimusmäärittely:	
Yleiset vaatimukset:	
Ulkonäkö suunnitellaan Tuusulan kunnan graafista ohjeistusta noudattaen. Sovelluksesta suunnitellaan mahdollisimman helppokäyttöinen. Käyttäjän tulee pystyä käyttämään sovellusta ilman käyttöohjeita.	
Tekniset vaatimukset:	
Sovellus kehitetään React Nativin avulla. Testauksen apuvälineenä käytetään Expo sovellusalustaa ja testauksen tulee onnistua testaajan omalla mobiililaitteella. Tietokannanhallintajärjestelmänä käytetään MongoDB Atlas tietokantaa.	
Toiminnalliset vaatimukset:	
Ensimmäinen vaihe:	
Etusivu, jossa näkyvät kaikki ehdotetut trennit.	Tiedot haetaan tietokannasta ja näytetään listalla etusivulla. Listan trenni avaa trennisivun.
Trennisivu, jossa voi tarkastella kyseistä trenniä tarkemmin.	Trennisivulla tulee näkyä treenin laji, kesto, lähtöaika ja päivä sekä lähtöpaikka.
Treenin lisääminen.	Käyttäjä voi lisätä uuden trenniehdotuksen ja siihen liittyvät tiedot muille käyttäjille.
Infosivu	Infosivulle lisätään linkki kyselylomakkeeseen testaamista varten.
Toinen vaihe:	
Kirjautuminen sovellukseen, joka sisältää kirjautumis- ja rekisteröitymissivut.	Käyttäjän tietojen tallentaminen tietokantaan, tunnuksien palauttaminen sähköpostin avulla. Kirjautunut käyttäjä pääsee etusivulle.
Trennisivu:	Lisätään osallistuminen trenniin ja osallistumisen poistamisen -toiminnot. Mikäli on ehdottanut trenniä, eikä muita osallistujia ole toiminto poistaa myös trenniehdotuksen. Tietojen poistaminen tietokannasta.
Karttatoiminto (Voidaan myös jättää pois lopullisesta toteutuksesta)	Käyttäjä voi asettaa lähtöpaikan kartalle treenin lisäyssivulla ja kartta tulee näkyviin trennisivulle. Tallentaminen koordinaattien avulla.
Julkaisuvaihe:	
Projektin julkaisu puhelimelle ladattavan linkin kautta tai julkaisu sovelluskaupassa. Päätetään yhdessä toimeksiantajan kanssa.	

Kuvio 5: Vaatimusmäärittely

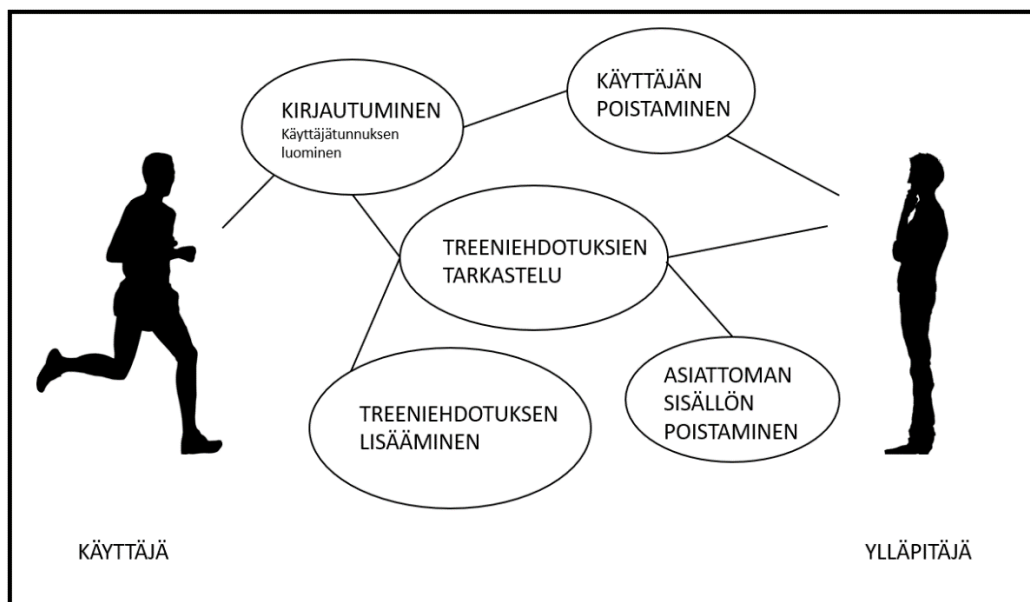
5.3 Käyttötapauskaavio

Käyttötapauskaavio on visuaalinen tapa tehdä vaatimusmäärittely. Käyttötapaus kuvaa prosessia ja soveltuu toiminnallisten vaatimusten määrittelyyn. Hyvässä käyttötapauskaaviossa esitetään visuaalisesti ja sanallisesti käyttäjän kommunikaatio ja toiminnot järjestelmän kanssa. (Kähönen 2016.)

Käyttäjä: Tuusulalainen. **Tavoite:** saada lenkkikaveri. **Laukaisija:** liikunnan tarve. **Käyttötapaoksen kulku:** Tuusulalainen kirjautuu sovellukseen, tarkastelee ehdotettuja treenejä, lisää uuden treenin.

Ylläpitäjä: Tuusulan kunnan liikuntatoimi. **Tavoite:** Estää epäasiallinen käyttö. **Laukaisija:** Ilmoitus / seurannassa esiin tullut epäasiallinen käyttö. **Käyttötapaoksen kulku:** Epäasiallinen treeniehdotus poistetaan, käyttäjä saa varoituksen / hänen käyttäjätunnuksensa poistetaan.

Kuviossa 6 olen kuvannut käyttäjän ja ylläpitäjän vuorovaikutusta sovelluksen kanssa esimerkin avulla.



Kuvio 6: Käyttötapauskaavio

6 Prototyypin suunnittelu

Suunnittelun ideointivaiheessa näkymiä ja toimintoja hahmoteltiin ensin luonnostelemalla. Luonnostelun etuna on muutosten tekemisen helppous ja rakenteen hahmottuminen alkuvaiheen suunnittelussa. Tarkempi ulkoasun suunnittelu tehtiin Figma-suunnitteluohjelman avulla. Työväline valittiin sen ilmaisen käytön ansiosta ja koska se oli tuttu aiemmista projekteista.

Mobiilisovelluksen suunnittelussa tulee erityisesti huomioida käyttäjän kulku näkymien välillä. Kirjautumisen tulee olla helppoa, sillä vaikka ihmiset selailevat puhelintaan tappaakseen aikaa, he eivät halua tuhlata aikaansa. Näkymien tulee olla selkeitä, informatiivisia ja kaikkien toimintojen tulee olla perusteltuja, monesti lisätiedot kannattaa sijoittaa erilliseen omalle sivulle. Painikkeiden tulee olla tarpeeksi suuria kulun helpottamiseksi. Myös tiedon hakemiseen ja tallentamiseen kuuluva latausaika tulee minimoida. (Nielsen, Budiu 2013, 68-85, 93, 120).

Yleisesti verkkosovelluksissa käytetään sivu-nimitystä ja mobiilisovelluksissa näkymä tai näyttö. Tässä työssä käytetään kuitenkin selkeyden vuoksi eri näkymistä myös nimitystä sivu.

6.1 Tunnistautuminen - uusi käyttäjä

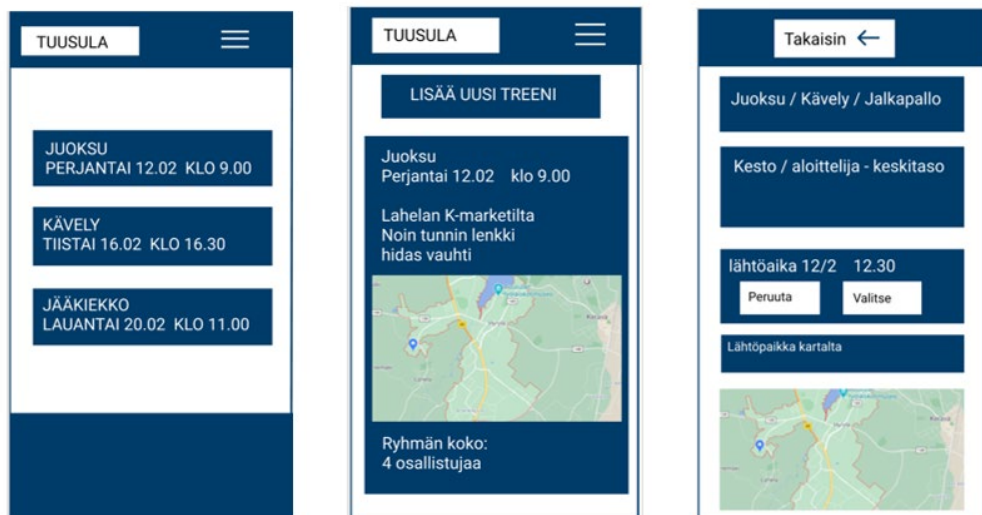
Kirjautuminen mahdollistaa käyttäjien määrän seuraamisen ja ehkäisee sovelluksen väärinkäyttöä. Lisäksi kirjautumisen avulla on mahdollista lisätä sovellukseen tulevaisuudessa uusia lisäominaisuuksia, esim. käyttäjän aktiivisuuden seurannan. Rekisteröidytessä palveluun, uudelta käyttäjältä kysytään käyttäjänimi ja sähköpostiosoite. Lisäksi käyttäjä luo itselleen salasanan. Kirjautunut käyttäjä ohjataan etusivulle. Kuviossa 7 on esitelty Figman avulla suunnitellut tunnistautumissivut.



Kuvio 7: Kirjautuminen ja rekisteröityminen

6.2 Treenisivut

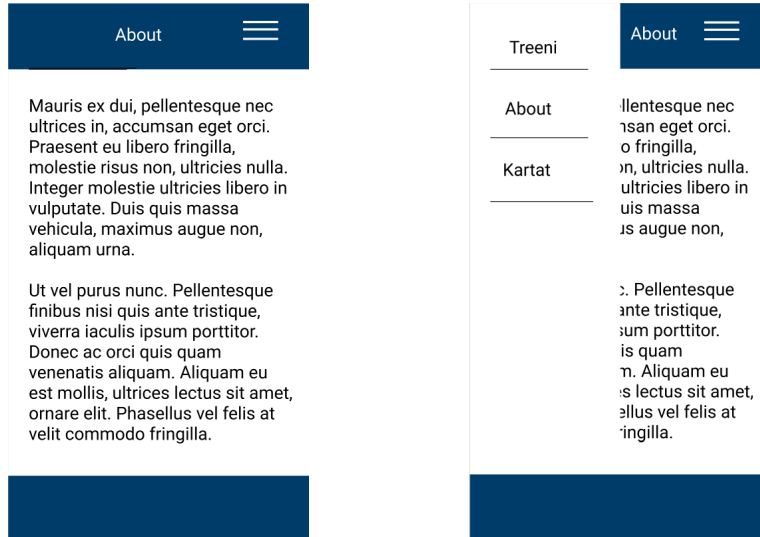
Käyttäjä ohjataan kirjautumisen jälkeen sovelluksen etusivulle. Etusivulla näkyy käyttäjien ehdottamat treenit aikajärjestyksessä. Etusivulta käyttäjä voi siirtyä tarkastelemaan treeniä tarkemmin tai lisäämään uuden treeniehdotuksen. Treenisivulla voi myös ilmoittautua mukaan kyseiseen treeniin. Kuviossa 8 näkyy Figman avulla tehty suunnitelma treenisivujen ulkoasusta ja toiminnoista.



Kuvio 8: Treenisivut

6.3 Infosivu

Yläosan menu-valikon kautta käyttäjä pääsee siirtymään infosivulle, jonne lisätään testin ajaksi linkki kyselylomakkeeseen. Valikkoon voidaan lisätä tulevaisuudessa uusia sivuja. Sivulla voidaan esimerkiksi näyttää käyttäjän oman aktiivisuuden seuranta tai muita lisäpalveluita kuten kävelyreittejä tai venyttelyohjeita. Kuviossa 9 näkyy Figman avulla tehty suunnitelma infosiivusta sekä yläosan menu-painikkeen kautta esiin tuleva valikkonäkymä.



Kuvio 9: Infosivu ja sivuvalikko

7 Työn toteutus

Työn tietolähteinä käytettiin pääasiassa React-, React Native- ja Expo-dokumentaatioita. Reactin perusteiden ja JavaScriptin osaaminen antoi hyvän pohjan kehittämiseksi. Paljon oppimista tapahtui myös kokeilujen ja erehtymisten kautta. Kirjalähteenä käytettiin lisäksi React Native in action -kirjaa (Dabit 2019).

Tässä kappaleessa kuvataan kehitystyön toteuttamisen osa-alueita kehitysympäristön rakentamisesta sovelluksen julkaisemiseen testausta varten. Pyrkimyksenä on esitellä juuri React Native-kehittämisessä tärkeitä osa-alueita, kuten navigaation kehittäminen ja eroavaisuuksien käsittely iOS- ja Android-käyttöjärjestelmien välillä.

7.1 React Native ja Expo

React Native -kehitysympäristö luodaan komentokehotteen avulla. Tässä työssä apuna käytettiin Expo CLI kehitysympäristöä, joka tarjoaa helpon tavan aloittaa tutustumisen React Native-kehittämiseen. Expo asennetaan koneelle Node.js-ajoympäristön avulla. Expo blank-rakenteen valinta antaa mahdollisuuden kehittää sovellus ns. puhtaalta pöydältä. Kuviossa 10 näkyy Training-projektin aloittaminen.

```
C:\Users\riikk>expo init Training
✓ Choose a template: » blank a minimal app as clean as an empty canvas
✓ Downloaded and extracted project files.

📦 Using Yarn to install packages. You can pass --npm to use npm instead.

✓ Installed JavaScript dependencies.

📦 Your project is ready!
```

Kuvio 10: Projektin luominen

7.2 React Native-tiedostorakenne

React Native-tiedostorakenne sisältää omat kansionsa iOS- ja Android-kehittämistä varten. Expo yhdistää nämä kansiot ja valitsee taustalla tarvittavat tiedostot käyttöjärjestelmän perusteella. Node-moduuli kansiossa sijaitsevat sovelluksessa käytetyt ja asennetut lisäosat, esimerkiksi tässä työssä käytetty datetimepicker-lisäosa. Käyttäjälle näkyvä tieto kulkee app.js-tiedoston kautta ja yhden sivun sovelluksessa riittääkin koodin kirjoittaminen pelkästään siihen.

Sovelluksen suunnittelussa kannattaa kiinnittää huomiota sovelluksessa tarvittavien tiedostojen kansiorakenteeseen. Selkeä rakenne auttaa sovelluksen hallinnassa. Mitä monimutkaisempi sovellus on kyseessä, sitä tärkeämpää on suunnitella sille selkeä rakenne. Hyvien käytänteiden mukaan kaikki sovellukseen kuuluvat tiedostot sijoitetaan yhden kansion sisään, jonne ne ryhmitellään omiin kansioihinsa toimintojen perusteella. Yhteys tietokantaan tehdään erillisen Server-kansion kautta. Server app.js muodostaa yhteyden ja määrittelee tietokannan rakenteen. Lisäksi määritellään tietokantaan tehtävät sovelluksen tarvitsemat haut ja viennit.

Kuviossa 11 on esitelty tämän työn tiedostorakenne.

Kuvio 11: Tiedostorakenne

7.3 Natiivikomponentit

Reactin web-komponenttien sijaan React Native käyttää rakennuspalikoinaan natiivikomponentteja. Tarvittavat komponentit kuten Text-, View-, Button- tai Image-komponentit tuodaan sovelluksen käyttöön React Native -alustalta. View-komponentti näyttää osion, jonka sisällä esim. teksti-komponentti sijaitsee. Mobiililaitteissa kätevän aktivointitoiminto TouchableOpacityn avulla voidaan aktivoida kohdekomponentti. (React Native 2021). Alla olevassa kuviossa 12 on esimerkki React Native -komponenttien käytöstä.



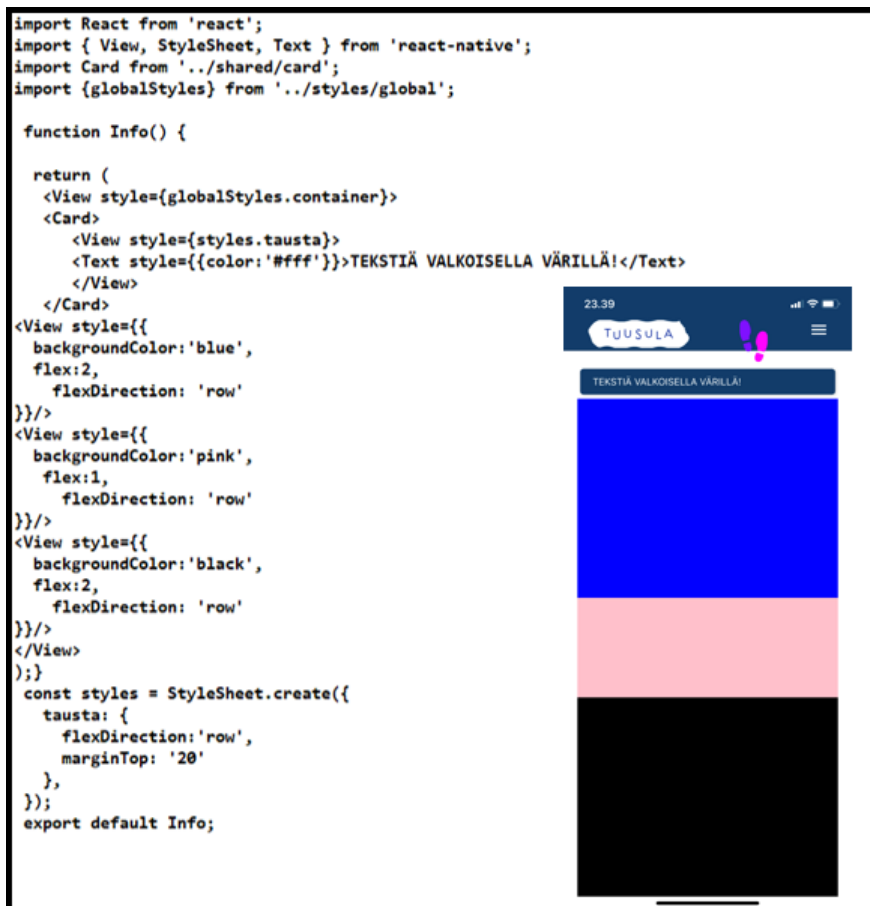
Kuvio 12: React Native-komponenttien käyttö ja näkymä iOS-puhelimessa

7.4 Sovelluksen ulkoasun muotoilu

Mobiililaitteiden näytön koko voi vaihdella aina älykellon näytöstä tabletin suuruiseksi. React Native tarjoaa mahdollisuuden määrittellä komponentin koon suhteessa näytön kokoon React Nativen Flexboxin avulla. Flexboxin avulla määritelty komponentti kasvaa dynaamisesti käytävissä olevan tilan mukaan. FlexDirection määrittelee komponenttien sijainnin suhteessa toisiin komponentteihin.

Mobiililaitteille suunniteltaessa tulee ottaa myös huomioda haluaako suunnitella erilaisen näkymän, mikäli mobiililaitetta käytetään pystysuunnan sijaan vaakasuunnassa. Käyttöjärjestelmille löytyy myös ohjeistus tummalle tai vaalealle näytön taustalle mukautuvan värimäärittelyn suhteen.

React Native tarjoaa erilaisia tapoja muokata ulkoasun tyyliä. Yhteisten tyylitiedostojen käyttäminen vähentää koodin määrää ja tuo sovellukselle yhtenevän ulkoasun. Tyyliä voidaan määrittellä myös sivukohtaisesti tai pelkästään tietyn koodirivin tasolla. Kuviossa 13 on esitelty tyylin määrittely kaikilla edellä esitellyillä tavoilla. View-komponentin tyyli on määritelty yhteisessä tyylitiedostossa, taustan tyyli määritellään sivukohtaisesti ja tekstin ja väriosioiden tyylit rivikohtaisesti.



Kuvio 13: Tyylitiedosto-esimerkki ja -näkyvä

7.5 Navigaatio

Navigaation, eli näkymästä toiseen liikkuminen on yksi mobiilisovelluksen ydintoiminnallisuuksista. Reitityksien ja navigaation suunnitteluun kannattaakin käyttää aikaa. React Nativen navigaatio poikkeaa verkkosivujen navigoinnista, koska käytössä ei ole sivun verkko-osoitetta vaan siirtyminen tehdään komponenttien välillä. Myös dataa eli informaatiota voidaan siirtää komponenttien välillä navigaation avulla. React Nativella alustalla ei ole käytössä omaa navigaatio

kirjastoon vaan on käytettävä kolmannen osapuolen kirjastoja, kuten React Navigaatiota. (Dabit 2019, 145-146).

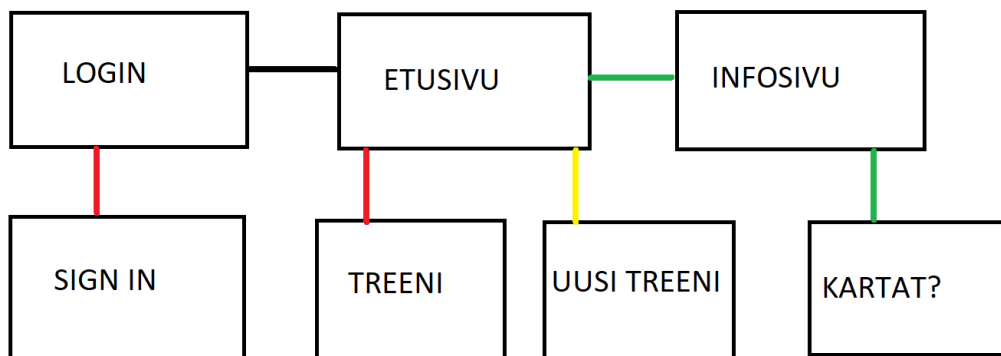
Stack-navigaatiossa näkymät siirtyvät toistensa päälle ja palattaessa edelliseen näkymään ne poistetaan yksitellen edellisen näkymän tieltä. Stack navigaatiossa liikkuminen tapahtuu aina seuraavan tai edellisen sivun kautta. Drawer-navigaatio toimii tyypillisesti sivusuunnassa esiin tulevan näkymävalikko listan avulla. Drawer-navigaatio korvaa edellisen sivun toisella sivulla, joten liikkuminen useampien sivujen välillä on ketterämpää. (Dabit 2019, 146).

Projektissa käytettiin kahta Stack-navigaatiota ja näitä yhdistävää Drawer-navigaatiota. Tässä työssä siirtyminen kirjautumissivulta rekisteröitymissivulle ja etusivulta treenisivulle on tehty Stack-navigaation avulla. Drawer-näkymässä näkyvät etusivu, infisivu sekä karttasivu. Siirtymisen uuden treenin lisäykseen tehdään moduulin kautta. Moduuli avautuu edellisen sivun päälle ja se voidaan avata ja sulkea painikkeen avulla.

Stack-navigaatio on merkitty kuvioon 14 punaisella värillä, Drawer vihreällä ja moduuli keltaisella värillä. Kirjautumistoimintojen Stack-navigaatiosta pääsee etusivulle kun kirjautuminen on hyväksytty. Tämä toiminto lisätään kokonaisuuteen testausvaiheen jälkeen.

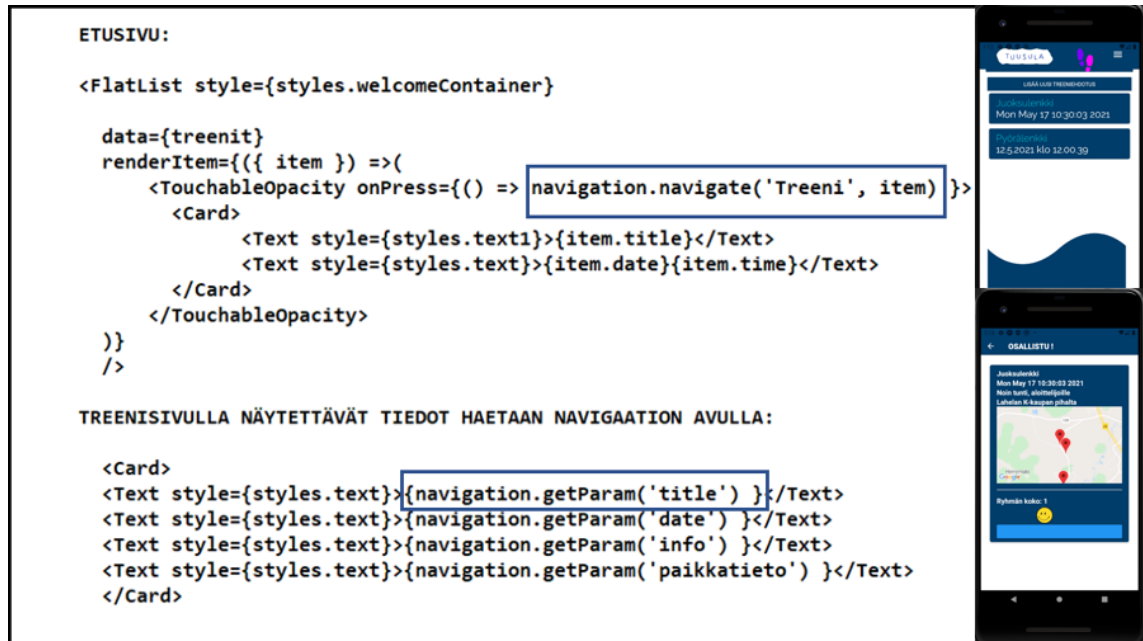
Navigaatio, eli näkymien välillä liikkuminen, suunniteltiin kahdessa osassa. Sisäänkirjautuminen ja käyttäjätunnusten luominen muodostavat ensimmäisen osan. Kirjautunut käyttäjä pääsee etusivun kautta siirtymään muihin näkymiin.

Etusivulta pääsee siirtymään treenisivulle, jolla on tarkempi kuvaus treenistä, tai moduuliin, jolla voi lisätä uuden treeniehdotuksen. Lisäksi luotiin navigaatio kotisivulta lisätietoja sisältäville sivuille. Tämä navigaatio mahdollistaa lisätoimintojen ja näkymien lisäämisen kehitystyön jatkokehitysvaiheessa.



Kuvio 14: Navigaation rakenne

Navigaation avulla siirrettiin myös tietoja näkymien välillä. Esimerkiksi kuviossa 15 treenimuoto ja -aika näytetään etusivulla ja painettaessa treeniä siirrytään treeni moduuliin ja samalla kyseisen treenin tiedot siirtyvät moduulin käyttöön. Treenisivulla harjoite näytetään yksityiskohdaisemmin käyttäjälle hakemalla navigaation avulla halutut tiedot.



Kuvio 15: Tietojenkäsittely navigaation avulla ja sen tuottamat näkymät Android-puhelimessa

7.6 Formik

Treenin lisäämiseen tarvittiin helppokäyttöinen lomake uuden treenin tietojen lisäämiseksi. Tässä työssä käytettiin lomakepohjana Formik-komponenttia, jonka valmiit toiminnot helpottavat tiedon käsittelyä. Formikin käyttö helpottaa lomakkeen käsittelyä valmiiksi rakennettujen metodiensa ansiosta.

Lomakkeen arvojen validointi tehtiin Yup:n avulla. Yup on JavaScriptin tiedon mallinnuksen rakentaja, joka auttaa lomakkeen arvojen validoinnissa. Arvoille määritellään nimi ja tiedon tallentamisessa käytetty muoto. Esimerkiksi teksti, eli String tai numero, eli Number. Yup:n avulla halutut kentät voidaan merkitä pakollisiksi. Mikäli pakollista kenttää ei olla lähettämävaiheessa täytetty, huomautusteksti näytetään käyttäjälle. Arvoille voidaan antaa Yup:n avulla monenlaisia kriteereitä, kuten annettujen merkkien minimi- tai maksimimäärä tai tiettyjen merkkien sisältyminen annettuun arvoon.

Käyttäjältä kysyttiin uuden treenin treenimuoto, esim. juoksu, treenin kesto ja taso, lähtöaika ja päivä sekä lähtöpaikka. Kaikki kentät määriteltiin pakollisiksi ja merkkien määräksi asetettiin minimissään 4 merkkiä. Päivyyriin asetettiin erikseen minimiarvoksi nykyhetki, koska harjoitteet on tarkoitettu asetettavaksi tulevaan ajankohtaan. Lähtöpaikan määrittämisen avuksi käyttäjälle näky kartta, johon on merkitty eri lähtöpaikkoja.

Tekstikenttien sisältämän tiedon käsittely Formikin avulla (kuvio 16).


```


const treeniSchema = yup.object({
  title: yup.string()
    .required('lisää treenimuoto')
    .min(4),
  info: yup.string('lisää lenkin kesto/ taso')
    .required()
    .min(4),
  date: yup.string().required('lisää lähtöaika'),
});

<View style={globalStyles.container}>
  <Formik
    initialValues={{ title: '', info: '', paikkatieto:'', date: new Date().toString()}}
    validationSchema={treeniSchema}
    onSubmit={(values, actions) => {
      actions.resetForm();
      addTreeni(values);
    }}
  >
  {(props) => (
    <TextInput
      style={globalStyles.input}
      placeholder='Juoksu / Kävely / Jalkapallopele?'
      placeholderTextColor = "#fff"
      onChangeText={props.handleChange('title')}
      value={props.values.title}
      onBlur={props.handleBlur('title')}
    />
    <Text style={globalStyles.errorText}>
      { props.touched.title && props.errors.title }</Text>
  )}
  </View>

```

Yup validaatio määrittelee onko kenttä pakollinen, virhetekstin käyttäjälle ja merkkien minimi määrän.

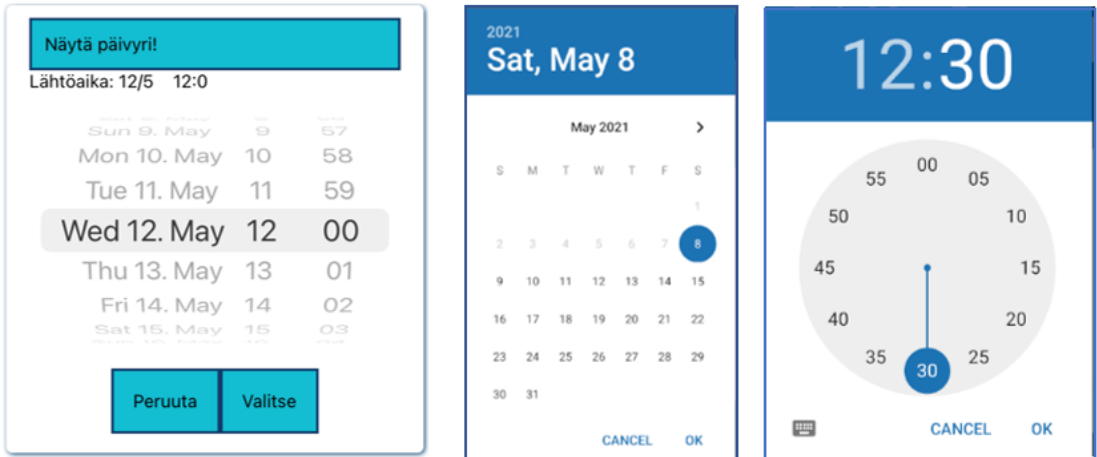
Arvojen alkuperäisarvot. Yup -validointi. Tietojen lisääminen addTreeni -arvoihin.



Kuvio 16: Tekstikenttien käsittely Formikin avulla

7.7 iOS- & Android-käyttöjärjestelmien eroavaisuuksien huomioiminen

Usealle eri puhelin käyttöjärjestelmälle kehitettäessä haasteena ovat eroavaisuudet eri järjestelmien välillä. Sama komponentti näyttää eri järjestelmässä erilaiselta, tai toimii eri tavalla. Esimerkkinä tässä työssä käytetty datetimepicker, eli päivän ja ajan valinta. iOS-järjestelmässä valitsimen avulla voitiin valita samanaikaisesti sekä päivä että aika, mutta Android järjestelmälle täytyi valinta tehdä erillisten päivä- ja aika-valitsimien kautta. Kuviossa 17 näkyy datetimepicker iOS- ja Android-laitteilla.



Kuvio 17: Päivän ja ajan valinta iOS- ja Android-erot

React Native tarjoaa eroavaisuuksien käsittelyyn kaksi erilaista tapaa. Mikäli kyseessä on pieni eroavaisuus voidaan käyttää Platform-moduulia, joka havaitsee kumpi käyttöjärjestelmä on

kulloinkin käytössä ja valitsee oikean vaihtoehdon. Kuviossa18 on yksinkertainen eroavaisuus käsitelty Platform-moduulin avulla.

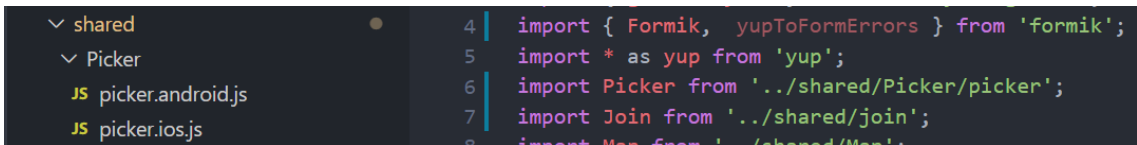
```
import { Platform } from 'react-native';

display = { Platform.OS === 'ios' ? "spinner" : "default" }

//näyttö mikäli kyseessä ios platform spinner, muuten oletus näyttö
```

Kuvio 18: Platform käyttö

Isommissa eroavaisuuksissa kannattaa kuitenkin erotella käyttöjärjestelmien koodit erillisiin tiedostoihin. Tiedoston nimi erotellaan .ios ja .android loppuosan avulla. React Native tunnistaa kummasta käyttöjärjestelmästä on kyse ja hakee oikean komponentin käyttöönsä. Kuviossa 19 on luotu erilliset kansiot IOS- ja Android-käyttöjärjestelmille. Käyttöön tuodaan Picker ja React Native tunnistaa käyttöjärjestelmän ja hakee oikean kansion.



```
4 | import { Formik, yupToFormErrors } from 'formik';
5 | import * as yup from 'yup';
6 | import Picker from '../shared/Picker/picker';
7 | import Join from '../shared/join';
8 | import Map from '../shared/Map';
```

Kuvio 19: Erillisten tiedostojen käyttö

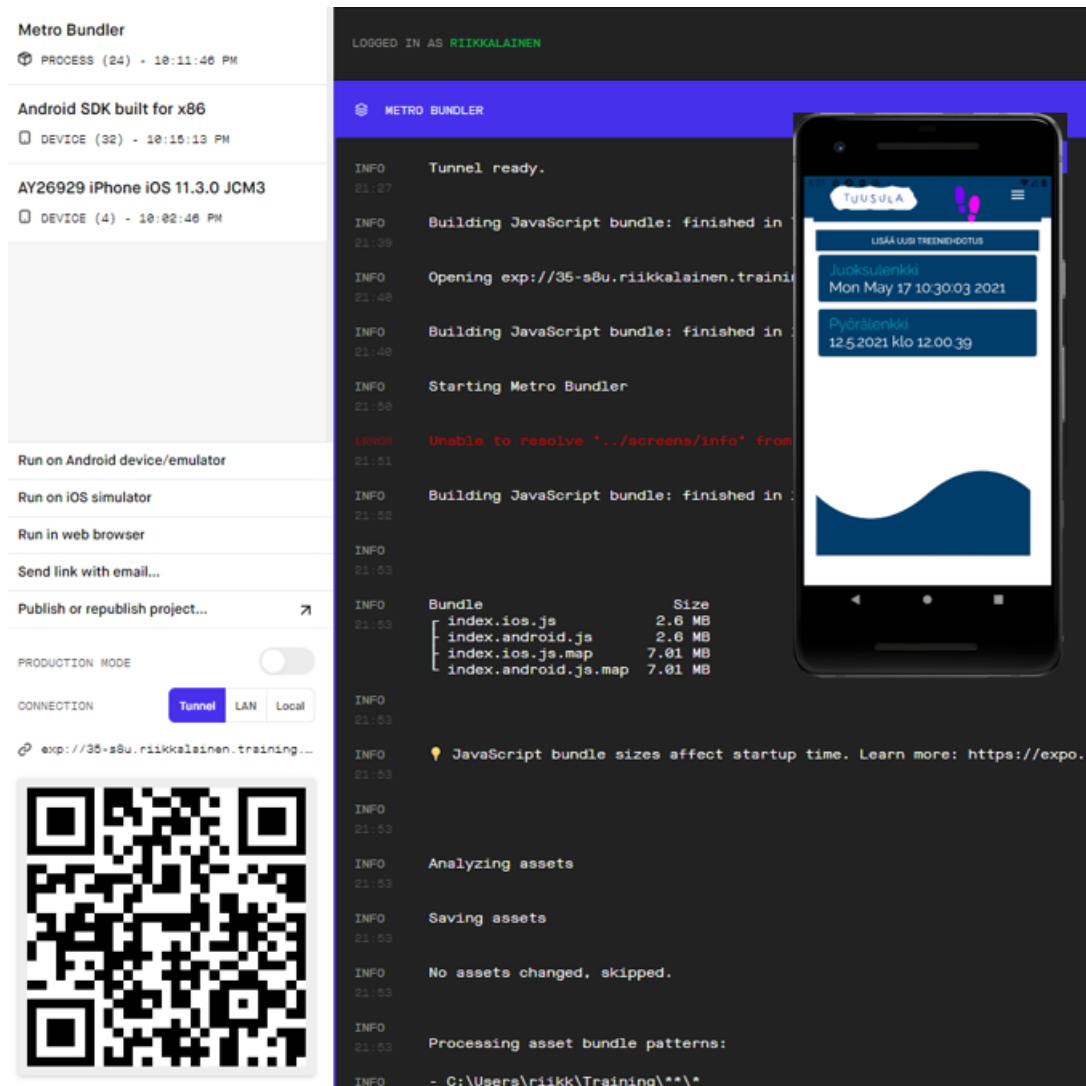
7.8 Tietojen tallentaminen ja hakeminen tietokannasta

Työssä käytettiin MongoDB Atlas-pilvipalvelua, jonka käyttö on yksinkertaisempaa, eikä sitä tarvitse asentaa tietokoneelle erikseen. Palveluun kirjautumisen jälkeen luotiin uusi tietokanta ilmaisen palvelun kautta valitsemalla palvelulle alueellinen palveluntarjoaja. Tietokannalle rekisteröitiin käyttäjä, jolle annettiin luku- ja kirjoitusoikeudet luotuun tietokantaan.

Dokumenttivaraston sisälle luotiin kokoelma (collection), johon tiedot tallennetaan avain - arvopareina. Yhteys tietokantaan luotiin sovelluksen kautta. Mongoosen avulla määriteltiin käsiteltävän tiedon rakenne (Schema) ja malli (Model). Rakenteessa määriteltiin tallennettavat tiedot ja niiden oletusarvot. Mallin avulla voidaan tehdä erilaisia kyselyitä tietokantaan. Tässä projektissa määriteltiin Treenitiedot Skeema (rakenne), joka pitää sisällään treenin nimen, lisätiedot, paikkatiedon ja kellonajan. Kyselyiden avulla haettiin kaikki treenitiedot ja tallennettiin uusi treenitieto. Lisäksi jatkokehitystä varten luotiin ”päivitä treenitieto” ja ”poista treenitieto” -kyselyt.

7.9 Prototyypin julkaisu Expon avulla

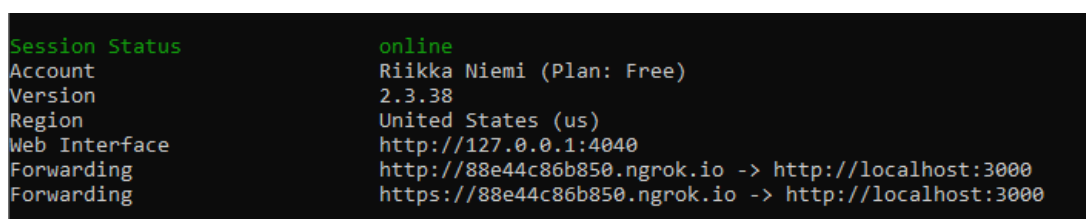
Expo CLI:n avulla kehitetty sovellus on helppo julkaista selainkäyttöliittymän avulla tai komentokehotteen avulla. Julkaisuun vaaditaan Expo-tiliin kirjautuminen. Julkaiseminen mahdollistaa sovelluksen reaaliaikaisen tarkastelun esim. tiimin kesken, tai tässä tapauksessa testiryhmälle. Expo CLI antaa käyttäjälle mahdollisuuden käyttää localhost-, LAN- ja tunnel-yhteyttä. Localhost ja LAN toimivat vain samassa verkossa olevilla laitteilla. Tunnel muodostaa suojatun yhteyden, jonka avulla käyttäjä saa yhteyden sovellukseen. Kuviossa 20 näkyy Expon Metro Bundler -käyttöliittymä ja esimerkkitalanne, jossa tarkastelen sovelluksen ulkoasua Android-simulaattorin avulla.



Kuvio 20: Expo kehitysalusta

Testausta varten oma paikallinen serveri täytyi jakaa ulkopuoliseen käyttöön turvallisesti. Tässä työssä yhteys muodostettiin Ngrok-sovelluksen avulla.

Käytännössä sovellus asennetaan tietokoneelle. Tämän jälkeen komentokehotteen avulla luodaan suojattu julkinen yhteys omaan paikalliseen palvelimeen. Kuviossa 21 näkyy Ngrok osoite ja paikallisen palvelimen yhteys.



Kuvio 21: Ngrok yhteys serveriin

8 Mobiilisovelluksen testaus

Testaaminen suoritettiin käyttäjätestauksena ns. mustalaatikko menetelmän mukaisesti, jolloin käyttäjillä ei ollut ennakkotietoa sovelluksen rakenteesta tai toiminnoista ennen

testaamista. Testauslaitteena toimi testaajan oma mobiililaitte. Testaajille annettiin kirjalliset ohjeet (liite 2) sovelluksen avaamista varten omalla mobiililaitteella. Ohjeet olivat joko tulostettuina (50 kpl), tai näkyvissä verkkosivulla. Testaaminen toteutettiin ilman valvontaa testaajan omassa ympäristössä. Testaamiseksi riitti sovelluksen avaaminen ja siihen tutustuminen. Testitulokset kerättiin testin jälkeen täytettävän kyselylomakkeen avulla. Testausmahdollisuutta tarjottiin kahden viikon ajan 15.4-30.4.2021 Tuusulan Lahelan alueella.

Testaukselle asetettiin kaksi tavoitetta: Ensimmäisenä tavoitteena oli selvittää prototyypin toiminta ja käytettävyys testin aikana. Toisena tavoitteena oli kartoittaa testaajien kiinnostusta sovelluksen käyttöä kohtaan ja selvittää olisiko palvelulle käyttäjiä Tuusulassa. Testaus käynnistettiin jakamalla esitteitä kohderyhmälle ja lisäämällä linkki Facebook-ryhmään. Liitteessä 1 on testausjoukon saatekirje ja liitteessä 2 on testin ohjeet.

8.1 Testiryhmä

Ennen varsinaista testaamista Expon toiminta ja kyselylomake pilotoitiin neljän esitestaajan avulla. Pilotoinnista kolme oli tietojenkäsittelyn opiskelijoita ja yksi työskentelee IT-alalla. Myös toimeksiantaja arvioi kyselylomaketta. Pilotoinnin etuna oli suu-sanallinen yhteys testaajaan, jolloin sovelluksesta saatiin muitakin havaintoja, kuin vain kyselylomakkeen kautta saatavat tiedot. Pilotoinnin ratkaisevan tärkeä havainto oli, että iOS-laitteella ei pystynyt avaamaan sovellusta QR-koodin avulla. Testaaminen onnistui kuitenkin kirjautumalla Expo-tilille. Ohjeistusta muutettiin tämän huomion vuoksi. Yksi pilotointiin osallistunut ehdotti avoimen palautteen lisäämistä sovelluksen käytön arviointia varten. Lisäksi toimeksiantaja toivoi avointa palautetta suoraan Tuusulan kunnalle. Nämä muutokset lisättiin kyselylomakkeeseen.

Testiryhmän haluttiin edustavan lopullista käyttäjäryhmää ja tämän johdosta testausmahdollisuutta tarjottiin Tuusulalaisille. Testi suoritettiin Tuusulan Lahelan alueella. Tulostettu esite jaettiin 50 talouteen alueella. Lisäksi tilaisuutta tarjottiin Lahelan Facebook-ryhmässä, jossa käyttäjiä on noin 1600 henkilöä. Ilmoitukseen liitettiin sähköinen linkki sivustolle, joka sisälsi samat tiedot ja ohjeet, kuin paperisessa esitteessä. Esitteessä kerrottiin kehitystyön tarkoituksesta ja tarjottiin mahdollisuutta päästä testaamaan ja antamaan kehitysideoita puhelimen sovelluksen jatkokehitystä varten.

Testaaminen perustui vapaaehtoisuuteen. Testaajan tuli ladata Expo Go -sovellus puhelimeensa ja skannata QR-koodi, jonka avulla sovelluksen sai auki, tai kirjautua sovellukseen testitunnusten avulla. Lisäksi testaajan tuli vastata liitteenä olevaan kyselyyn.

8.2 Testikysymykset

Testikysymysten tarkoituksena oli selvittää prototyypin käytettävyyttä ja kiinnostusta sovellusta kohtaan. Lisäksi testaajilta kysyttiin kehitysehdotuksia jatkokehitystä varten. Testaamiseksi riitti, että testaaja oli avannut sovelluksen ja selaillut sovellusta.

Testikysymykset voidaan jakaa kolmeen osaan: Vastaajan taustatietoihin, sovelluksen käyttöhalukkuuden selvittämiseen ja sovelluksen käytettävyyttä koskeviin kysymyksiin.

Testaajan taustatiedoista kysyttiin vastaajan ikä ja puhelinkäyttöliittymä. Lisäksi kartoitettiin testaajan nykyisiä liikuntatottumuksia. Testaaja arvioi kuinka usein hän harrastaa liikuntaa viikoittain ja liikkuuko hän mieluiten yksin vai ryhmässä.

Testaajien kiinnostusta sovelluksen käyttöä kohtaan selvitettiin kysymällä Likertin-arvoasteikon avulla arviota siitä, käyttäisivätkö he itse vastaavanlaista sovellusta. Lisäksi testaajilta kysyttiin kyllä-ei-vastauksen avulla uskoivatko he, että sovellukselle olisi käyttäjiä Tuusulassa.

Sovelluksen käytettävyyttä selvitettiin kysymällä testaajilta miten he testasivat sovellusta. Vaihtoehtoina olivat esimerkiksi selaileminen, treeniehtotuksen lisääminen. Lisäksi kysyttiin vastaajan arvio sovelluksen käytön vaikeudesta ja oliko harjoitteen lisääminen helppoa.

Kyselylomakkeessa kysyttiin myös testaajan arvio siitä, aktivoiko sovellus liikkumaan. Avoimen palautteen kautta pyydettiin palautetta sovelluksen käytön onnistumisesta ja testaajan

omia havaintoja sekä kehitysideoita jatkokehitystä varten. Lisäksi tarjottiin mahdollisuutta antaa avoimen palautteen kautta ehdotus Tuusulan vapaa-aikapalveluille.

8.3 Tulokset

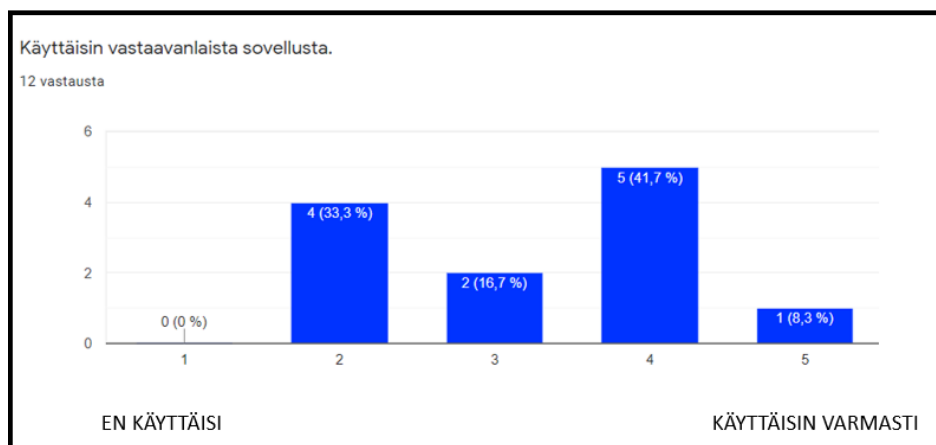
Testin kyselytutkimukseen vastasi yhteensä 13 henkilöä. Näin ollen otoskoko tutkimuksessa oli N=13. Koehenkilöitä oli mukana lähes kaikista ikäryhmistä. Hieman yli puolet vastaajista 7 henkilöä oli 40-50-vuotiaita. Ikäryhmistä ainoastaan yli 50-vuotiaita ei ollut testaajien joukossa. Palautetta saatiin tasaisesti sekä iOS että Android käyttäjiltä. Lähes puolet testaajista oli Android käyttäjiä ja hieman yli puolet iOS käyttäjiä. Testaajista kolme selaili sovellusta ja kahdeksan ehdotti treeniä.

Vastanneista lähes kaikki (12 henkilöä) arvioivat harrastavansa liikuntaa kolme kertaa viikossa tai useammin. Enemmän hajontaa ilmeni siitä, liikkuko mieluummin ryhmässä vai yksin (kuvio 22).



Kuvio 22: Liikkuminen yksin vai ryhmässä

Kakki testaajat (13 henkilöä) arvioivat, että sovellukselle olisi käyttäjiä Tuusulassa. Kysymyksen aktivoiko sovellus liikkumaan 3 vastasi ei, 4 valitsi neutraalin vastauksen ja 6 uskoi, että sovellus aktivoi liikkumaan. Arvio käyttäisikö itse sovellusta vastauksissa oli samankaltaista hajontaa (kuvio 23).



Kuvio 23: Sovelluksen käyttö

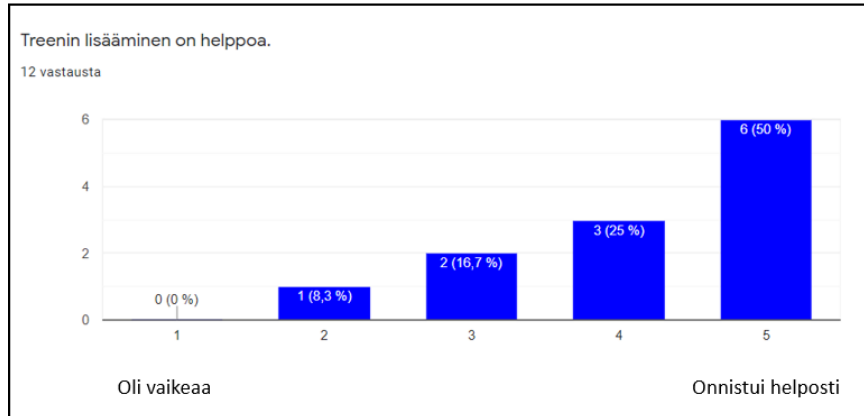
Avoimia kommentteja käytön arvioinnille oli melko vähän:

”Pidin ideasta, ja yksinkertaisuudessaan eihän se tuon monimutkaisempi tarvitse olla”.

”Perusidea on sinällään minusta hyvä ja sovelluksen käyttäminen onnistui lennosta”.

”Oletan, että tästä tulee varsin hyvä ja käytettävä sovellus”.

Kahdeksan vastaajaa arvioi, että sovellusta oli helppo käyttää. Neutraaleja ”ei helppo eikä vaikea” -vastauksia saatiin kolmelta vastaajalta. Treenin lisäämisen helppouden arviointi antoi samankaltaisen tuloksen ja koettiin arvioinneissa (kuvio 24) enemmän helpoksi, kuin vaikeaksi.



Kuvio 24: Treenin lisäys

Avoimen palautteen perusteella saatiin selvyttä käytössä ilmenneisiin ongelmiin:

”Appi ei toiminut”.

”Pääsin hyvin selailemaan ehdotettuja treenejä. Liittyminen ei onnistunut”.

”Toimiiko paikan ilmoittaminen kartalta”?

”Kentissä, joihin kirjoitetaan treenin tyyppi, kesto/taso on vaikea nähdä aktiivi. Voisin ehdottaa lisäämään pienet otsikot kenttiin”.

”Aloittelija / keskitaso -valinnat ja kestot varmaan toimisivat paremmin rullausvalikkona”?

”Kellonaikojen sekuntitieto on turhaa tietoa”.

8.4 Esiin tulleet haasteet

Expo CLI-alustan yhteysongelmien takia kaksi testaajista ei onnistunut avaamaan sovellusta. Mobiilitestaaminen omalla laitteella asettaa omat haasteensa testaamiselle. Yhteys saattaa katketa testin aikana. Syynä tähän voi olla testausalustan kuormittuminen tai oman palvelimen internetyhteyden katkeaminen.

Helppokäyttöisiä testiympäristöjä ei ole tarjolla ja vastausprosentti jää helposti matalaksi testaamisen hankaluuden takia. Testin tekemiseen Expon avulla vaaditaan sovelluksen lataamista omalle puhelimelle. Puhelimen muistitila on rajallinen ja tämä voi olla syynä haluttomuuteen osallistua testiin. Lisäksi testaaminen vaatii esimerkiksi QR-koodin skannaustaitoa tai kirjautumista testiympäristöön. Sovelluksen testaaminen vaatii testaajalta vaivannäköä ja kiinnostusta testin tekemiseen. Testaajan täytyi ladata puhelimeensa Expo Go-sovellus. Tämän jälkeen Android-käyttäjät pääsivät avaamaan sovelluksen QR-koodin avulla ja IOS-käyttäjät joutuivat kirjautumaan sovellukseen. Kyselyyn pääsi vastaamaan sovelluksen info-sivun linkin kautta.

Todennäköisin syy alhaiseen vastausprosenttiin oli testaamisen vaikeus. Ohjeistusta olisi voinut tehdä selkeämmäksi. Testaajia motivoitiin kahdelle testaajalle arvotun palkinnon avulla. Kaksi testaajista voitti veneen käyttöönsä Tuusulanjärvelle päiväksi retkieväineen. Testaamiseen olisi voinut motivoida enemmän kaikille osallistuneille jaettava palkkio.

8.5 Testitulosten arviointi

Tutkimuksen laatua voidaan mitata rehabiliteetin ja validiteetin perusteella. Rehabiliteetti arvioi tutkimustulosten pysyvyyttä ja tarkkuutta, mikäli tutkimus toistettaisiin. Validiteetin avulla arvioidaan mittasiko testi sitä mitä sen avulla oli tarkoitus selvittää. Lisäksi validiteetti kertoo vastaako tulos todellisuutta ja olisiko tulos yleistettävissä. (Middleton 2020.)

Testituloksia voidaan arvioida määrällisin keinoin, numeraalisesti ja tilastollisilla menetelmiä käyttäen. Määrällinen tutkimus edellyttää suurempaa testitulosten määrää ja tuottaa yleistettävissä olevaa tietoa aiheesta. Laadullisen arvioinnin kautta etsitään tuloksista merkityksiä sanallisessa muodossa ja sitä käytetään esimerkiksi ilmiöiden ymmärtämiseen. Laadullisen tutkimustuloksen saavuttamiseen riittää pienempi testiryhmä. (Streefkerk 2021.)

Testin tavoitteena oli selvittää sekä prototyypin toimintojen käytettävyyttä, että kiinnostusta käyttää sovellusta tulevaisuudessa.

Testin tavoite selvittää testiajien kiinnostusta käyttää sovellusta, ei tämän testauksen aikana toteutunut. Testi olisi kannattanut suunnitella kaksiosaisena, jolloin kiinnostusta käytölle olisi voitu selvittää määrällisen tutkimuksen avulla, laajemmalla kyselyllä. Tällöin tuloksen validiteetti olisi ollut parempi ja tarjonnut paremmin yleistettävissä olevaa tietoa kiinnostuksesta käyttää sovellusta. Tämän testin tulos kertoo vain kolmentoista henkilön kiinnostuksesta sovelluksen käyttöä kohtaan. Testiryhmä edustaa asuinpaikkakuntansa puolesta loppukäyttäjien ryhmää. Testaajiksi valikoitui kuitenkin teknisesti osaavampaa ja liikuntatottumuksiltaan aktiivisempaa joukkoa, eikä tulos ole yleistettävissä Tuusulan kuntalaisiin yleisesti. Mikäli tämän tutkimuksen tuloksia halutaan kuitenkin tulkita halukkuudessa käyttää sovellusta, voidaan todeta, että juuri tähän tutkimukseen osallistuneista puolet olisivat halukkaita itse käyttämään sovellusta ja kaikki uskoivat, että sovellukselle olisi käyttäjiä Tuusulassa.

Käytettävyyden selvittämisessä tutkimuksen tavoite saavutettiin tässä testissä. Moranin (2019) mukaan käytettävyyden arvioimiseksi jo viiden testaajan tulos riittää selvittämään suurimman osan käytettävyyden ongelmista. Vastauksia saatiin myös melko tasaisesti sekä Android-, että iOS-käyttöliittymien käyttäjiltä, joten tulosten perustella voidaan tehdä päätelmiä molempien käyttöliittymien osalta. Mobiilisovelluskehityksessä käyttäjätutkimus omalla puhelimella tarjoaa mahdollisuuden havainnoida käyttöä todellisessa ympäristössä. Tämä antaa realistisen kuvan sovelluksen käytöstä ja ympäristössä olevat häiriötekijät ja verkkoyhteyden laatu vastaavat normaalia tilannetta mobiililaitetta käytettäessä.

Valvontaa tai ohjeistusta testin tekemiseen ei tarjottu testaajille. Tämä poistaa testaajan vaikutuksen testaustulokseen ja antaa realistisemman kuvan sovelluksen helppokäyttöisyydestä. Ohjeita sovelluksen käytölle ei haluttu antaa, koska sovellus oli suunniteltu yksinkertaisuutta silmällä pitäen ja toimintoja tulisi osata käyttää ilman erillistä ohjeistusta. Tuloksista voidaan päätellä, että sovellusta oli melko vaivatonta käyttää ilman ohjeistusta. Myös treenin lisäämien osoittautui helpoksi.

Osa testaajista olisi halunnut liittyä treeniin, tai käyttää karttatoimintoja testauksen yhteydessä. Olisikin ollut järkevää kertoa selkeästi, että nämä toiminnot tullaan lisäämään sovellukseen myöhemmin. Myös treenin poistamis-toimintoa toivottiin palautteen perusteella. Toisaalta halukkuus käyttää edellä mainittuja toimintoja osoittaa niiden tarpeellisuuden jatkokehittämisen aikana. Lisäksi avoimen palautteen kautta saatiin hyviä korjaus- ja kehittämiskohteita jatkokehittämiseen.

Tulosten perusteella voidaan todeta sovelluksen toimivuus. Testin perusteella sovellus oli myös helppokäyttöinen ja harjoitteen lisääminen onnistui helposti. Tulokseen kuitenkin vaikuttaa oletus, että testiryhmäksi valikoitui teknisesti keskivertoa osaavampi ryhmä. Käytettävyyttä olisi hyvä selvittää myös vähemmän teknisesti osaavan joukon kesken, jolloin saataisiin luotettavampi arvio käytettävyydestä.

8.6 Arviointi toimeksiantajan palautteen perusteella

Tutkimustulokset esitettiin toimeksiantajalle tutkimuksen päätyttyä. Lisäksi esitettiin arvio sovelluksen jatkokehityksen vaiheista ja aikataulusta sekä mahdollisista julkaisuvaihtoehdoista ja -kustannuksista. Toimeksiantaja arvio tuloksia mielenkiintoisiksi, mutta totesi, että lisäperehtymistä aiheeseen vielä tarvitaan. Hänen mukaansa sovellus kannattaisi ottaa käyttöön osana hyvinvointiprojektia, jolloin projektiin osallistujille tarjottaisiin sovelluksen avulla mahdollisuutta sopia yhteisiä aktiviteetteja. Sovellus tarjoaisi hyvinvointiteemaan uuden lisäpalvelun.

Sovelluksen tarjoaminen rajatun ryhmän käyttöön on mielestäni todella hyvä ajatus. Käyttäjien tuntiessa toisensa, olisi helpompi sopia ja osallistua aktiviteetteihin. Osallistujat voisivat tarjota toisilleen vertaistukea ja kannustusta hyvinvointiteeman tavoitteissa. Lisäpalvelu voisi myös mahdollistaa osallistujien ryhmäkoon kasvattamisen, koska osa aktiviteeteista tapahtuisi omaehtoisesti ryhmän kesken. Teeman mukaan sovellukseen voisi lisätä myös siihen sopivia lisäominaisuuksia. Aktiivisuuden seuraaminen ryhmän jäsenien kesken voisi kannustaa liikkumaan enemmän. Oman aktiivisuuden seuraamisen lisäksi osallistuja voisi mahdollisesti seurata omaa aktiviteettiaan muhin ryhmän jäseniin verrattuna.

Lisäperehtymistä vaaditaan vielä, esimerkiksi sovelluksen käytön turvallisuuden ja mahdollisten vastuukysymysten osalta. Myös julkaisualustaa koskeviin kysymyksiin ja kustannuslaskelmaan liittyvät kysymykset tulee selvittää.

9 Sovelluksen jatkokehittäminen

Testitulosten perusteella voitiin määritellä tärkeimmät jatkokehityskohteet. Jatkokehityksessä sovellukseen lisätään kirjautuminen. Kirjautumisen avulla voidaan vähentää väärinkäytöksiä ja saada yhteys sovelluksen käyttäjiin. Kirjautumisen yhteydessä on mielekästä lisätä myös treeniin liittyminen ja oman treeniehdotuksen poistaminen, sillä toiminnot yhdistetään kirjautuneen henkilön profiiliin. Myös palautteen perusteella ilmoittautuminen treenin koettiin tärkeäksi toiminnallisuudeksi. Tämä tullaan lisäämään kirjautumisen yhteydessä. Samalla lisätään myös mahdollisuus poistaa oma treeniehdotus, mikäli siihen ei ole vielä muita ilmoittautuneita osallistujia.

Lisäksi havaittiin ongelmia Android-käyttöliittymän toiminnoissa, jotka tulee korjata jatkokehityksessä. Android-käyttöliittymän alavalikko näkyy ”lisää treeni”-painikkeen päällä. Jatkokehityksessä suunnitellaan ulkoasu tarkemmin Androidille sopivaksi, tarkastellaan toiminnot läpi ja tehdään mahdolliset korjaukset Android-puhelimen avulla.

Palautetta saatiin myös päivä- ja aika-valinnoissa. Androidin päivä- ja kellonaika-toiminnot suunnitellaan uudella tavalla. Valinnoissa varmistetaan, että päivämäärä näytetään käyttäjälle vain kerran. Päivä ja aika muotoa korjataan: Päivä / kuukausi, tunnit / minuutit -muotoon.

Mikäli kartta tulee lopulliseen sovellukseen, tulee, siinä olla mahdollisuus merkitä lähtöpaikka kartalle. Kartta sijoitetaan omalle sivulle, että sitä on helpompi käyttää.

Treenin taso- ja kesto -valinnat toteutetaan valmiiden vaihtoehtojen pohjalta valittaviksi. Tämä oli uusi kehittämisidea testikäyttäjältä avoimen palautteen kautta. Ennalta määritelty taso on helpompi asettaa ja treeniin osallistujat tietävät minkä tasoinen treeni on kyseessä.

Kehittämistyön lopuksi testataan koodin toiminta ja korjataan tarvittavat virheet. Julkaisu voidaan tehdä Expon avulla. Ohjeistuksena käytetään Expo Building Standalone Apps-dokumenttiota. Sovellus toteutetaan sekä iOS- että Android-käyttöliittymille. Sovelluksen voi tämän jälkeen julkaista verkkosivulta suoraan puhelimeen ladattavana linkkinä, tai tehdä julkaisu Apple Store- ja Google Play-kauppoihin. Kustannuksissa tulee huomioida, että MongoDB, Google karttapalveluiden sekä Expon tilit on tehty opiskelijoille suunnatuilla ilmaisversioilla. Tietokannan käytön hinnoittelu perustuu käytön mukaan laskutettavaan malliin ja Google-

karttojen hinnoitteluun vaikuttavat käytettävän kartan ominaisuudet ja toiminnot. Lisäksi sovelluksen julkaisu on maksullista. Kehitystyön julkaisulle tulee varata aikaa, sillä esimerkiksi Apple Storella on tarkat vaatimukset, jotka sovelluksen tulee täyttää.

Kehittämistyö tullaan tekemään loppuun. Julkaisu riippuu toimeksiantajan tarpeesta sovellukselle ja kustannusten hyväksymisestä kunnassa. Realistinen aikataulu ajoittuu syyskuun alkuun 2021.

10 Pohdinta

Oppinäytetyössä kehitettiin toimiva prototyyppi React Nativen avulla. Evoluutio-prototyyppi toimii hyvänä lähtökohtana jatkokehitystä varten. Lisäksi tutkimuksessa saadun palautteen perusteella saatiin testaajilta käytännöllisiä jatkokehitysideoita prototyypin jatkokehittämistä varten.

Testituloksien perusteella sovelluksen käytölle ilmeni kiinnostusta. Vastausmäärä jäi kuitenkin liian pieneksi todellisen kiinnostuksen kiistattomaksi toteamiseksi. Mikäli aito kiinnostus sovellusta kohtaan halutaan selvittää, ehdottaisin kyselytutkimuksen tekemistä Tuusulan kunnan asukkaiden keskuudessa. Käyttäjätestauksen yhteydessä esitetty kysely vaatii testaajalta vaivannäköä ja siksi vastaustulos jää väistämättä liian pieneksi, jotta käytön halukkuus voitaisiin todentaa.

React Native oli uusi, mielenkiintoinen ja opettavainen teknologia. Käytännön projektityössä oppiminen on mielekästä ja antaa itseluottamusta tulevaisuuteen. Ongelmien selvittämisen kautta oppiminen oli työssä antoisinta ja tarjosi sekä epätoivon, että riemun hetkiä. Sovelluskehitysprojektin tekeminen itsenäisesti alusta loppuun antaa todella hyvän kokonaiskuvan prosessista ja kehittäjänä pääsee käyttämään sekä luovuuttaan suunnittelussa ja aiemmin opittuja taitoja koodipuolella. Osaamisen kannalta React Native-kehitys oli hyvä jatkumo JavaScript- ja React-osaamiseen. Toki näissä edellä mainituissakin opittavaa vielä riittää.

Kehittäminen Android- ja iOS-käyttöliittymälle samanaikaisesti, toi kokemattomalle uudenlaisia haasteita. Ihastuin iOS käyttöliittymän tarjoamaan mahdollisuuteen valita päivämäärä ja kellonaika samanaikaisesti rulla-valikosta. Käytin paljon turhaa aikaa Android sovelluksen saamiseksi samankaltaiseen muotoon, kun valitsemalla ensin päivämäärä- ja tämän jälkeen kellonaika-toiminnoilla olisi saatu toimivampi ratkaisu Androidille. Toinen huomionarvoinen asia on, että kehitin Android-sovellusta pääasiassa simulaattorin avulla ja iOS-sovellusta omalla puhelimella. Tästä syystä toiminnallisuuksien testaaminen ei ollut yhtä sujuvaa Android puhelimelle.

Testaamisessa työn laajuuden rajaamiseksi jouduttiin tyytymään kompromissiin, jossa haluttiin todentaa ehkä liiankin monta asiaa. Toimeksiantajan intressinä oli selvittää kiinnostus sovelluksen käyttöä kohtaan ja tuotteen kannalta taas haluttiin selvittää prototyypin toimivuus. Lisäksi oma kokemattomuus testaamisessa näkyi testituloksissa. Käyttäjätestauksesta oppi tämän työn kautta paljon ja nyt valmiudet testauksen tekoon olisivat lähtökohdiltaan huomattavasti paremmat.

Sovelluksen idea on mielestäni ajankohtainen. Ihmiset muuttavat yhä enemmän töiden perässä uusille paikkakunnille ja naapureihin on vaikeampi tutustua kuin aikaisemmin. Lenkille lähteminen tai yhteisen pesäpallon pelin järjestäminen antaa osallistujilleen enemmän kuin liikkuminen itsessään. Facebookin ryhmiin liittyminen saattaa olla hankalaa, mutta sovelluksen avulla treenien sopimisen kynnyksistä helpottuu.

Toimeksiantajan ehdotus tarjota sovellusta tietyn ryhmän käyttöön, oli mielestäni erinomainen. Kynnyksistä lähtee tuntemattoman henkilön kanssa lenkille voi olla liian suuri, mutta toimisi erinomaisesti tutun ryhmän kesken.

Lähteet

Painetut

- Barnum, C. M. & Krug, S. 2011. *Usability testing essentials: Ready, set..test*. Amsterdam: Elsevier / Morgan Kaufmann.
- Boduch, A. 2020. *React and React Native*. 3. painos. Packt Publishing.
- Dabit, N. 2019. *React native in action*. Shelter Island, NY: Manning Publications.
- Haikala, I. & Märijärvi, J. 2006. *Ohjelmistotuotanto*. 10. painos. Helsinki: Talentum.
- Nielsen, J. & Budiu, R. 2013. *Mobile usability*. Berkeley, Calif.: New Riders.
- Rodriguez Martinez, E. 2018. *React: Cross-platform application development with React Native : build 4 real-world apps with React Native*. Packt Publishing.
- Juvonen, R. 2018. *Ohjelmistoprojektin sudenkuopat ja miten ne vältetään*. Helsinki: Books on demand.

Sähköiset

- Botelho, B. 2021. Definition MongoDB. Viitattu: 18.03.2021. <https://searchdatamanagement.techtarget.com/definition/MongoDB>
- Bracey, K. 2018. What is Figma? Envato Tuts+, 26 November. Viitattu: 16.3.2021. <https://webdesign.tutsplus.com/articles/what-is-figma--cms-32272>
- Dawson, C. 2009. *Projects in Computing and Information Systems: A guide for students*. E-kirja. Pearson Education UK.
- Dooley, J. 2011. *Software Development and Professional Practice*. E-kirja. Dordrecht: Springer.
- Foster, E. G. 2016. *Database Systems*. Apress. 3-8/2016.
- Karnik, N. 2018. Introduction to Mongoose for MongoDB. Viitattu 28.4.2021. <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>
- Hall, J. 2015. Figma Introduces World's First Browser-Based, Professional Grade Tool for Collaborative Interface Design. Business Wire, 12/2015.
- Hamza, Z., & Hammad, M. 2020. Testing Approaches for Web and Mobile Applications: An Overview. *International Journal of Computing and Digital Systems*, 9(4). <http://journal.uob.edu.bh/handle/123456789/3877>
- Jabangwe, R., Edison, H. & Duc, A. N. 2018. Software engineering process models for mobile app development: A systematic literature review. *The Journal of Systems & Software*, 145/2018, 98-111.
- Kähönen, P. 2016. Perinteinen vaatimusmäärittely vs. käyttötapaukset. Viitattu 16.3.2021. <https://www.pasaati.com/blog/perinteinen-vaatimusm%C3%A4%C3%A4rittely-vs.-k%C3%A4ytt%C3%B6tapaukset>
- Liu, S. 2020. Cross-platform mobile framework used by software developers worldwide in 2019 and 2020. Statista, 2 July. Viitattu: 16.3.2021. <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- Middleton, F, 2020. Reability vs validity: what's the difference? Viitattu 5.5.2021. <https://www.scribbr.com/methodology/reliability-vs-validity/>

Münch, J. & Armbrust, O. 2012. *Software Process Definition and Management*. E-kirja. Springer Berlin Heidelberg.

Myers, G. J., Badgett, T. & Sandler, C. 2012. *Art of Software Testing*. E-kirja. John Wiley & Sons, Inc.

Mikko Mäntyneva. 2016. *Hallittu projekti: Jäntevästä suunnittelusta menestykselliseen toteutukseen*. E-kirja. Helsingin seudun kauppakamari.

O'Reilly. 2021. Learning React Native. Viitattu: 27.4.2021. <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>

Perea, P & Giner, P. 2017. *UX Design for Mobile*. E-kirja. Bringham: Packt Publishing.

Santanen, J. 2017. Tarvekartoitus ja vaatimusten analyysi. Viitattu 5.5.2021. <http://www.mit.jyu.fi/palvelut/sovellusprojektit/luennot/TarvekartoitusVaatimustenAnalyysi2s.pdf>

Sengupta, D., Corvalan, D. & Singhal, M. 2016. *Getting started with React*. E-kirja. Packt Publishing.

Seppälä, K. Tietotekniikan termitalkoot. <https://www.kielikello.fi/-/tietotekniikan-termitalkoot>

Stevens, E. 2020. Should You Learn JavaScript? Advice For Newbie Web Developers. Viitattu: 22.3.2021. <https://careerfoundry.com/en/blog/web-development/should-you-learn-javascript/>

Stuckey, M. I., Carter, S. W. & Knight, E. 2017. The role of smartphones in encouraging physical activity in adults. *International journal of general medicine*, 10/2017, 293.

Streefkerk, R. 2021. Qualitative vs. quantitative research. Viitattu 5.5.2021. <https://www.scribbr.com/methodology/qualitative-quantitative-research/>

Technopedia. Definition- What does Mobile Application mean? Viitattu: 19.5.2020. <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>.

Tilastokeskus. Suomen virallinen tilasto (SVT): Väestön tieto- ja viestintäteknikan käyttö. ISSN=2341-8699. Helsinki: Viitattu: 12.02.2021.

Tuusulan kunta. 2021. Vapaa-ajan palvelut. Viitattu 12.2.2021 https://www.tuusula.fi/sivu.tmpl?sivu_id=1042

Tuusulan kunnan verkkosivut. 2021. Viitattu 12.02.2021. <https://www.tuusula.fi/>

Expo dokumentaatio. 2021. Viitattu 02.03.2021. <https://docs.expo.io/>

Julkaisemattomat

Sähköpostikirjeenvaihto Vapaa-aikapalveluiden päällikkö Risto Kanerva. Tuusulan kunnan liikuntatoimi 2021. 10.05.2021.

Kuviot	
Kuvio 1: Ohjelmistoprojektin vaiheet	11
Kuvio 2: Evoluutiivisen kehittämisen malli	12
Kuvio 3: Figma-suunnittelutyökalu	16
Kuvio 4: Projektin aikataulu	16
Kuvio 5: Vaatimusmäärittely	17
Kuvio 6: Käyttötapauskaavio	18
Kuvio 7: Kirjautuminen ja rekisteröityminen	19
Kuvio 8: Treenisivut	19
Kuvio 9: Infosivu ja sivuvalikko	20
Kuvio 10: Projektin luominen	20
Kuvio 11: Tiedostorakenne	21
Kuvio 12: React Native-komponenttien käyttö ja näkymä iOS-puhelimessa	21
Kuvio 13: Tyylitiedosto-esimerkki ja -näkymä.....	22
Kuvio 14: Navigaation rakenne	23
Kuvio 15: Tietojenkäsittely navigaation avulla ja sen tuottamat näkymät Android-puhelimessa	24
Kuvio 16: Tekstikenttien käsittely Formikin avulla	25
Kuvio 17: Päivän ja ajan valinta iOS- ja Android-erot	25
Kuvio 18: Platform käyttö	26
Kuvio 19: Erillisten tiedostojen käyttö	26
Kuvio 20: Expo kehitysalusta.....	27
Kuvio 21: Ngrok yhteys serveriin	27
Kuvio 22: Liikkuminen yksin vai ryhmässä.....	29
Kuvio 23: Sovelluksen käyttö.....	29
Kuvio 24: Treenin lisäys.....	30

Liitteet

Liite 1: Saatekirje kyselyyn	38
Liite 2: Ohjeet testaukselle	39

Liite 1: Saatekirje kyselyyn

Treenikaveri kevään lenkeille?

Hei!

Lopputyönäni (tietojenkäsittelijä amk Laurea) olen kehittänyt prototyypin puhelinosovelluksesta Tuusulan kunnan liikunta- ja vapaa-ajanpalveluiden käyttöön.

Tavoitteena oli esitellä uudenlainen palvelu kuntalaisten käyttöön. Mahdollisuus kerätä pyöräily-, kävely- tai lenkkeilyporukka kasaan ja tavata uusia ihmisiä. Ehkäpä myös treenata yhdessä kahvakuulilla, pelata jalka- tai pesäpalloa?

Prototyyppi on valmis ja nyt tarvitaan sinun mielipidettäsi sovelluksesta. Tätä sovellusta testataan Tuusulan Lahelan alueella 15.4 – 30.4.2021.

Testaukseen riittää sovelluksen avaaminen ja toimintojen kokeilu, mutta koronarajoitusten puitteissa voitte myös osallistua ja ehdottaa oikeita lenkkejä muille testaajille. Toistaiseksi osallistujien määrä ei näy treenitiedoissa. Sovellus päivittyy, kun se avataan uudelleen.

Sovelluksen toiminnassa saattaa olla katkoksia testin aikana, joten toivon teiltä kärsivällisyyttä.

Lopullisessa versiossa tulee olemaan kirjautuminen sähköpostin avulla. Kirjautuminen mahdollistaa esim. asiattomien käyttäjien poistamisen. Lisäksi osallistujien määrä tulee näkymään kyseisen treenin lisätiedoissa.

Kokeilemalla sovellusta pääset mukaan kehittämään uudenlaista palvelua ja antamaan mielipiteesi palvelun hyödyllisyydestä. Kaksi onnekasta kyselyyn vastannutta voittaa soutuveneen käyttöönsä päiväksi Tuusulanjärvelle retkieväineen.

Testikäyttäjän tulee olla yli 18-vuotias. Vastauslinkin sähköpostiosoitteita tai puhelinnumeroa käytetään ainoastaan palkinnosta ilmoittamiseen ja tiedot hävitetään asianmukaisesti heti testin päätyttyä.

Ohjeet myös verkkosivullani:

<https://justfeelingfab.com/>

Vastauslinkki löytyy myös sovelluksen Info-sivulta:

<https://forms.gle/nJexMTkoYaChkPn17>

Kiitos osallistumisestasi sovelluksen testaukseen!

Riikka Niemi

riikka.niemi@student.laurea.fi

Nähdään lenkeillä?

Liite 2: Ohjeet testaukselle

Lataa puhelimeesi ilmainen **Expo go** sovellus.

(App Store / Google Play)



Android skannaa oheinen QR-koodi puhelimesi (esim. kameran avulla):



Avaa apissa Expo.

IOS / i-phone

Kirjaudu testiprofiiliin:

Valitse **Expo go** sovelluksen alaosan valikosta **Profile**, sininen **Sing in to your account** -vaihtoehto.

Username: riikkalainen

Password: testaja01

Avaa training sovellus.

MUKAVAA TESTAILUJAKSOA!

Vastauslinkki: (Löytyy myös sovelluksen *Info-sivulta*).

<https://forms.gle/nJexMTkoYaChkPn17>

Mikäli **QR-koodin luku kameralla** ei onnistu:

Ios >asetukset >kamera -skannaa QR-koodit päälle.

Android >avaa kamera >lisää >asetukset -aktivoi Google-Lens -ehdotukset.