

Ville Sorkkila

Automatisoitu drooni osana turva-alueprototyyppiä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinööriytyö

3.5.2018

Tekijä Otsikko	Ville Sorkkila Automatisoitu droni osana turva-alueprototyyppiä
Sivumäärä Aika	44 sivua 3.5.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	Smart Systems
Ohjaaja	yliopettaja Antti Piironen
<p>Insinööriyön tavoitteena oli toteuttaa Digital Wellbeing Sprintissä vuonna 2017 kehitetty konsepti Lapinjärven kunnalle. Konseptin pohjalta luotiin prototyyppi, jonka tarkoituksena oli havainnollistaa valvovaa järjestelmää, jonka avulla muistihäiriöisen henkilön turvallisuutta voitaisiin mahdollisesti parantaa, kun henkilö liikkuu maastossa. Järjestelmän tarkoituksena oli lisäksi helpottaa hoitohenkilökunnan päivittäisiä toimia.</p> <p>Toteutusvaiheessa Android-käyttöjärjestelmällä varustetulle mobiililaitteelle luotiin sovellus, jonka avulla kehitettiin valvontaan käytetyn dronin lento-ominaisuudet. Hoitohenkilökunnan toimien helpottamiseksi kaikki lento-ominaisuudet automatisoitiin. Lisäksi mobiililaitteelle luotiin käyttöliittymä, josta on nähtävissä lennätysten aikainen kulku, mikä välitetään käyttäjälle droniin kuuluvasta kamerasta. Dronia käytettiin henkilön kunnan tarkistamisessa palvelimelta vastaanotetun aktivointikäskyn jälkeen. Lennätys edellytti, että aluksen alkuperäinen sijainti on saatu onnistuneesti tallennettua ja palvelimelta vastaanotetut henkilön koordinaatit ovat käyttökelpoiset. Lennätysten aikana myös dronin sijaintitiedot välitettiin palvelimelle. Sijaintitiedot eri laitteilta kerättiin GPS:n avulla. Tiedonvälityskkeinona henkilön, palvelimen ja aluksen välillä hyödynnettiin CloudMQTT-palvelua. Turvakeinona manuaalisen ohjauksen palauttamiseen käytettiin dronin kanssa hankittua kauko-ohjainta.</p> <p>Prototyypin toinen kehittäjä loi henkilön paikannusjärjestelmän, joka koostui mobiililaitteesta ja palvelimesta. Palvelin vastaanottaa ja välittää viestejä eri osapuolille ja näyttää myös osapuolilta saatujen sijaintitietojen perusteella liikkeitä Google Maps -palvelua hyödyntävällä sivustolla. Sivustolla näytetään lisäksi käyttäjän määrittelemä turva-alue. Rajanylityksen tapahduttua henkilölle välitetään myös viestejä palvelimelta käyttämällä Telegram Bot -palvelua.</p> <p>Kehitetyn prototyypin avulla valvovan järjestelmän toteutusta pystyttiin havainnollistamaan rajoitetuin lentoetäisyyksin.</p>	
Avainsanat	droni, automatisoitu lennätys, Android, paikannusjärjestelmä, mobiililaitte, GPS

Author Title	Ville Sorkkila Automated drone as part of a safe area prototype
Number of Pages Date	44 pages 3 May 2017
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Smart Systems
Instructor	Antti Piironen, Principal Lecturer
<p>The goal of the thesis was to implement a concept developed in the Digital Wellbeing Sprint for the municipality of Lapinjärvi in 2017. Based on the concept, a prototype was developed to demonstrate a supervisory system which could help in improving the safety of the people with memory impairment when they are moving in terrain. Relieving personnel's daily workload was also one of the system's usage purposes.</p> <p>In the implementation phase drone's flight features were developed with an application that was created for an Android mobile device. To relieve personnel's workload, all flight features were automated. In addition, a user interface was created for the mobile device so that live video feed could be received from the drone's camera when flying between two destination points. Drone was used to check the person's condition after receiving an activation command from the server. To be able to fly, the aircraft's original location needed to be successfully recorded and the person's coordinates received from the server needed to be executable. Also, during flight, the drone's location coordinates were forwarded for the server. GPS was used to collect the location data from different devices. CloudMQTT service was used as a means of communication between the person, the server and the drone. A remote controller acquired with the drone acted as the safety feature so that manual control could be regained.</p> <p>Other developer of the prototype created the person's positioning system which consisted of a mobile device and a server. Messages are received and forwarded by the server for different parties and, based on the location data received from the parties, the movement is displayed in a website that uses Google Maps service. The user specified safe area is also displayed in the website. In addition, server transmits messages via Telegram Bot service for the person after he/she has crossed the safe area.</p> <p>Based on the developed prototype, a supervisory system could be demonstrated with limited flight distances.</p>	
Keywords	drone, Android, mobile device, positioning system, automated flight, GPS

Sisällys

1	Johdanto	1
2	Miehittämättömät ilma-alukset	1
2.1	Miehittämättömien ilma-alusten historia	2
2.2	Droonin valinta	5
2.3	Valitun droonin ominaisuudet	6
2.4	Lennättämiseen liittyvät määräykset Suomessa ja Euroopassa	10
3	Droonin käyttöönotto	13
3.1	Lennätyksen ja järjestelmien alustava testaus	13
3.2	Droonin toiminnallisuuksien avaaminen mobiiliympäristöön	14
3.3	Mobiilisovelluksen ja droonin yhdistäminen	18
3.4	Kameran käyttöönotto	21
3.5	Tiedonsiirto palvelimen ja mobiilisovelluksen välillä	24
3.6	Lentämiseen liittyvät toiminnot	28
3.7	Lopputulos	36
4	Huomioitavaa jatkokehitystä ajatellen	38
5	Yhteenveto	40
	Lähteet	42

1 Johdanto

Insinööriyön tarkoituksena on kehittää prototyyppi järjestelmästä, jonka avulla muistihäiriöisen asukkaan turvallisuutta voitaisiin parantaa ja samalla helpottaa hoitohenkilökunnan toimintaa, kun asukas liikkuu maastossa. Insinööriprojekti koostuu kolmesta aihealueesta: henkilöstä ja henkilön mukana olevasta mobiililaitteesta, eri toimintoja valvovasta palvelimesta ja miehittämättömästä ilma-aluksesta (dronista), joista tässä raportissa perehdytään erityisesti dronin toimintaan. Projektin toinen tekijä, opiskelija Santeri Qvintus kertoo insinööriyössään perusteellisemmin valvovasta palvelimesta ja henkilön mukana olevan mobiililaitteen eri toiminnoista. Insinööriyöraportissa käsitellään Digital Wellbeing Sprintissä vuonna 2017 Lapinjärven kunnalle kehitetyn konseptin toteutusta.

Raportissa käsiteltäviin asioihin sisältyy ilma-alusten historiaa ja tuotteen valintaan vaikuttavia kriteereitä ja syitä, miksi projektin toteutuksessa päädyttiin käyttämään DJI:n tuotetta. Valitun tuotteen toimintoihin, järjestelmiin ja teknisiin osiin perehdytään raportissa, ja käyttöönottamiseen sisältyvät vaiheet käsitellään niiden toteutusjärjestyksessä. Lisäksi paneudutaan projektin kehityksen aikana ilmenneisiin ongelmiin. Mahdollisia prototyyppin kehitykseen vaikuttavia rajoituksia tutkitaan myös Suomen ja Euroopan unionin lennättämiseen liittyvien määräyksien kautta.

Työn jatkokehityksen kannalta tutkitaan prototyyppin toteutuksen aikana havaittuja rajoittavia tekijöitä, kuten sääolosuhteita, toimintaympäristöä ja dronin teknisiä osia, jotka tulisi huomioida tarkasti, mikäli projektia lähdetään kehittämään esimerkiksi tuotantovaiheeseen asti.

2 Miehittämättömät ilma-alukset

Taustatyön avulla selvitetään, onko olemassa prototyyppin kehityksen kannalta rajoittavia tekijöitä, jotka estäisivät työn toteutuksen. Lisäksi historian kautta tutustutaan miehittämättömien ilma-alusten lähtökohtiin, käyttötarkoituksiin ja vuosien varrella tapahtuneeseen kehitykseen. Yleiskuvan muodostamiseksi perehdytään myös projektissa käytettävän dronin järjestelmiin ja pääpiirteisiin, samoin dronin hankintaan vaikuttaviin tekijöihin.

2.1 Miehitämättömien ilma-alusten historia

Miehitämätön ilma-alus juontaa juurensa 1800-luvulle saakka. Ensimmäiset miehitämättömät ilma-alukset kehitettiin puolustusvoimien käyttöön Itävallassa, kun se hyökkäsi Italian kaupunkiin Venetsiaan. Ensimmäinen versio olikin miehitämätön ilmapallo räjähteillä varusteltuna vuonna 1849 (1). Kuvassa 1 havainnollistetaan hyökkäystä Venetsiaan miehitämättömillä ilmapalloilla.



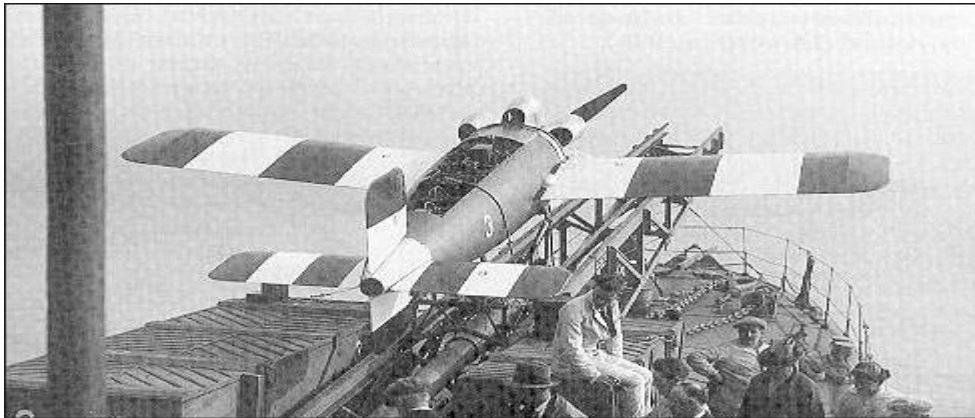
Kuva 1. Kauhupommitusten hyökkäys miehitämättömillä ilmapalloilla Venetsiassa vuonna 1849 (2).

Ensimmäisen maailmansodan loppuvaiheilla ilmaantuivat miehitämättömät lentokoneet. Ensimmäisten joukossa lentäjiä kehittämässä oli Sperry Gyroscope Companyn perustaja Elmer Sperry, joka loi aikaisen version nykypäivänä käytettävistä ilmatorpedoista nimeltä Hewitt Sperry Automated Airplane. (1.) Kuvassa 2 on nähtävissä Sperryn kehittämä versio ilmatorpedosta.



Kuva 2. Hewitt-Sperry Automatic Airplane vuonna 1918 (3).

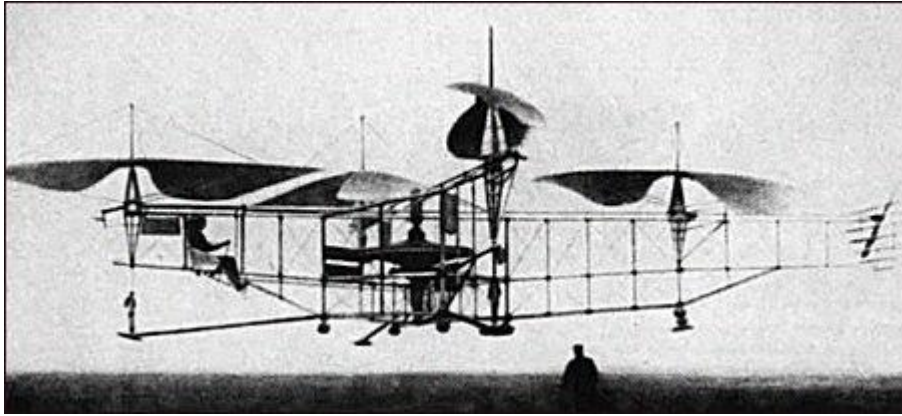
Sodan jälkeen Yhdysvaltojen armeija muutti standardit E-1-koneet miehittämättömiksi ilma-aluksiksi, joihin myös Larynx lukeutui. Larynx oli pienikokoinen yksitaso, joka sotaluoksesta liikkeelle laukaistua lensi autopilotilla. Lisäksi useita automaattisia ilma-aluksia kehitettiin vielä tämän jälkeen Yhdysvaltojen ja Ison-Britannian armeijan toimesta. (1.) Kuvassa 3 näkyy Larynx sen laukaisualustalla.



Kuva 3. RAE Larynx laukaisualustalla HMS Strongholdilla. Laatikon päällä on tohtori George Gardner, RAE:n tuleva johtaja. (4.)

Toisen maailmansodan aikoihin Reginald Denny, joka oli myös tunnettu näyttelijä Hollywoodista, kehitti konseptin ensimmäisille massatuotannossa oleville lentokoneille. Radio-ohjattavat miehittämättömät ilma-alukset olivat Dennyn mielenkiinnon kohde, jota varten hän myös kehitti yrityksen nimeltä Reginald Denny Company. Yrityksessä tuotettiin radio-ohjattavia koneita, joita Yhdysvaltojen armeija koekäytti toisen maailmansodan aikana ja joista armeija itse kehitti myös erityyppisiä ilmatorpedon versioita. Kylmän sodan aikana näitä aluksia käytettiin maalitauluina, ja ne pystyivät keräämään myös aktiivista radiodataa. (1.)

Quadrokopteri, tai tarkemmin sanottuna neliroottorinen kopteri, oli ensimmäisten joukossa kehitettyjä pystysuunnassa nousevia ja laskevia ilma-aluksia. Aikaisemmat helikopterimallit olivat peräroottorilla ja yhdellä pääroottorilla varusteltuja, minkä todettiin olevan epätaloudellista ja tehotonta. Helikopterin pystysuuntaisen liikkeen ongelmien ratkaisemiseksi insinöörit kehittivät neliroottoriset kopterit. Vuonna 1920 Etienne Omnichen loi ensimmäisen Omnichen 2 -quadrokopterin, joka lensi yli 1 000 onnistunutta lentoa ja ennätyselliset 360 metriä kerralla. (1.) Tämä quadrokopteri on kuvassa 4.



Kuva 4. Oehmichen numero 2 -quadrokooperi (5).

Ilma-alusten kehitys 1900-luvun alkupuolelta aina nykypäivään saakka on ollut valtaisa, varsinkin viimeisen 10 vuoden osalta. Sellaiset yritykset kuin Heli-Max, Blade, Walker, Parrot ja DJI ovat tuottaneet uusimmilla tietokoneteknologioilla varusteltuja aluksia hallittavuutta ja ilmakehän kuvauksia varten. Ominaisuuksien ja lähes päivittäin ilmestyvien uusien toiminnallisuuden takia miehittämättömät ilma-alukset ja quadrokooperit ovat erittäin suosittuja harrastelijoiden parissa. Kuvassa 5 on esimerkki nykypäivänä markkinoilla olevista drooneista.



Kuva 5. Bolt-drooni (6).

Vaikka ilma-alusten eri käyttötarkoitukset nykypäivänä ovat jo moninaiset, kehittävät yritykset silti niille jatkuvasti uusia tehtäviä, kuten metsäpalstojen ja ympäristön kuvaamista ja kartoittamista. Tulevaisuudessa alusten toivotaan esimerkiksi pystyvän kuljettamaan lääkkeitä ja ruokaa alueille, jotka ovat muuten saavuttamattomissa. (1.)

2.2 Droonin valinta

Kriteereinä insinööriprojektin hankittavalle droonille olivat seuraavat asiat: Droonia täytyy pystyä ohjelmoimaan, jotta erilaisia lentoa liittyviä toimintoja voidaan automatisoida halutulla tavalla. Laitteen hinnan tulee olla kohtuullinen, koska drooni hankitaan omakustanteisesti. Lentoajan tulee olla riittävä prototyypin testaamista varten. Laitteessa on oltava kamera, tai siihen täytyy olla mahdollista asentaa kamera, jotta päämäärässä olevaa henkilöä voidaan kuvata. Lisäksi laitteen mukana on oltava jokin hätämekanismi, jonka avulla drooni saadaan automatisoidusta lennätyksestä takaisin käyttäjän hallintaan vikatilanteen aikana (esimerkiksi kauko-ohjain).

Suunnitteluvaiheessa pohdittiin myös ratkaisua, jossa drooni rakennettaisiin itse, jolloin laitteen hankintaan liittyviä kustannuksia saataisiin alennettua. Tutkimisen jälkeen tultiin kuitenkin siihen tulokseen, että projektityöhön varattu aika ei tulisi riittämään, koska työsnettäviä osa-alueita (laitteen tasainen lennättäminen, ohjaukseen liittyvät toiminnot, moottoreiden tehon säätäminen erilaisten antureiden lukemien perusteella jne.) olisi liikaa. Jäljelle jäi valmiin sellaisen droonin valinta eri laitevalmistajien väliltä, joka täyttää laitteen hankinnalle asetetut kriteerit.

Droonin valinta osoittautui kuitenkin haastavaksi tehtäväksi, sillä aikaisempaa kokemusta aihealueesta ei ollut eikä tarkalleen ottaen tiedetty, kuinka hyvin eri laitevalmistajien tuotteilla automatisoituja toimintoja olisi mahdollista tehdä. Valinnassa epävarmuutta aiheutti myös se, että droonin hankinta projektiin tehtäisiin omakustanteisesti. Ratkaisu ongelmaan kuitenkin löydettiin erään Metropolian opettajan kanssa käydyn keskustelun jälkeen: hänellä oli aikaisempaa kokemusta droneista. Opettajalta saatiin erittäin hyviä neuvoja siitä, mitä eri laitteilla on mahdollista tehdä, erityisesti DJI-nimisen laitevalmistajan droneilla. Lisäksi saatiin hyödyllistä ja tärkeää tietoa siitä, mitä tulisi itse lennätyksen aikana ottaa huomioon, kuten esimerkiksi Suomessa ja Euroopassa lennätykseen liittyvät määräykset.

Saadun tiedon perusteella pystyttiin luomaan lähtökohta, josta prototyypin kehitys voitaisiin aloittaa. Laittevalmistajan drooneihin perehtymisen jälkeen päädyttiin hankkimaan DJI Spark -niminen drooni, sillä se täytti myös projektille asetetut kriteerit. Prototyypin toteutusta varten perehdyttiin myös lennättämiseen liittyviin määräyksiin, jotta saataisiin muodostettua kokonaiskuva siitä, mitä määräysten puitteissa on valitulla laitteella mahdollista toteuttaa.

2.3 Valitun droonin ominaisuudet

Projektiin valitulle droonille on mahdollista kehittää mobiilisovelluksia, joiden avulla droonin toimintoja voidaan automatisoida, hinta on kohtuullinen, laitteessa on kamera ja lentoaika on riittävä toimintojen havainnollistamista varten. Kuvassa 6 on nähtävissä prototyypin toteutuksessa käytettävä drooni.



Kuva 6. DJI Spark -drooni (7).

DJI:n drooneilla on tuki mobiilisovellusten luomiseksi joko Android- tai iOS-ympäristössä. Ohjelmoitavat toiminnot eri alustoille ovat nähtävissä laitevalmistajan sivuilla. Droonin toimintoihin ja teknisiin tietoihin tutustuttiin entuudestaan ennen varsinaisen laitteen hankintaa, jotta saataisiin muodostettua arvio sen ominaisuuksista. Kuvassa 6 näkyvää laitetta on saatavilla esimerkiksi maahantuojalta kahtena eri pakettina: normaalipakettina, johon sisältyvät perustarvikkeet itse droonin lisäksi, kuten akku, laturi, kantolaatikko ja microUSB-kaapeli, tai Fly More Combo -pakettina, jolloin mukana tulevat edellä mainittujen lisäksi vara-akku, lataustelakka, kaksi varapropellia, suojat droonin propelleille ja kauko-ohjain.

Kauko-ohjain

Normaali- ja Fly More Combo -paketeista päädyttiin jälkimmäisen hankintaan nimenomaan paketin mukana tulevan kauko-ohjaimen takia. Ohjain toimii Wi-Fi-yhteyspisteenä, johon droni ja mobiililaitte muodostavat yhteyden (8). Ohjaimessa olevan langattoman yhteyden avulla dronin lentoetäisyyttä lähtöpaikan ja kohdepaikan välillä pystytään kasvattamaan (valmistajan laitteistodokumentaation mukaan DJI Sparkin lentoetäisyys Euroopassa on 500 metriä ja ilman ohjainta 100 metriä). Ohjain toimii myös vikaturvana, mikäli automatisoidun lentämisen aikana tapahtuu jotain odottamatonta, kuten esimerkiksi GPS-signaali häviää, jolloin laite saatetaan joutua ohjaamaan manuaalisesti takaisin lähtöpisteeseensä. Dronin kanssa käytettävä kauko-ohjain on nähtävissä kuvassa 7.

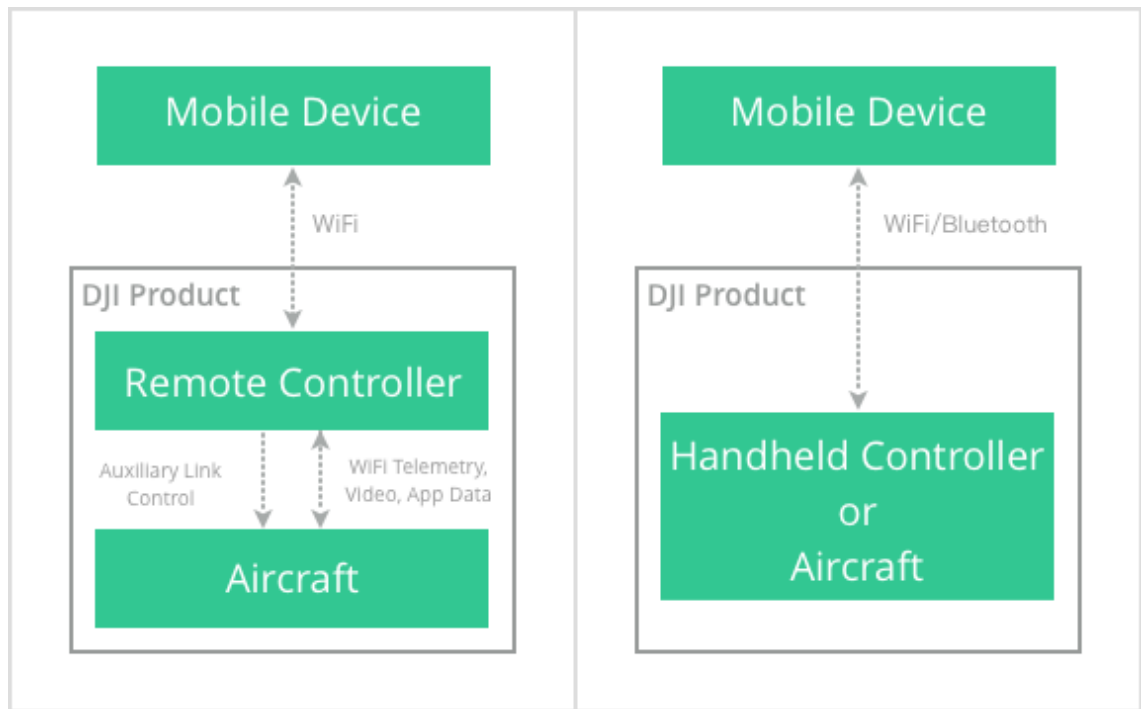


Kuva 7. DJI Sparkin kauko-ohjain (9).

Kauko-ohjaimessa olevasta kytkimestä on valittavissa kaksi lentotilaa, joissa laite voi toimia. P-tilassa (Positioning) droni käyttää kaikkia paikantamiseen tarkoitettuja apukeinoja, joihin kuuluu GPS, visiointijärjestelmä ja esteidenväistämisjärjestelmä. P-tila mahdollistaa automatisoidun lentämisen, kehittyneiden ominaisuuksien käytön ja toimii olustilana projektin toteutuksessa. Myös S-tilassa (Sport) ovat käytössä kaikki paikantamiseen tarkoitettut apukeinot, ja ohjattavuutta tehostetaan säätämällä käsiteltävyyden vahvistusarvoja. Esteidenväistämisjärjestelmä ei ole käytössä tässä tilassa, ja dronin maksiminopeus nousee 20 m:iin/s (maksiminopeus projektissa käytettävällä laitteella on 13,9 m/s laitteistodokumentaation mukaan). S-tilaa voidaan myös käyttää manuaalisen ohjauksen palauttamiseen automatisoidusta tilasta. (10.)

Yhteyden muodostaminen laitteiden välillä

Mobiilisovellus kehitettiin käyttämällä DJI:n mobiiliohjelmistokehityspakettia (Mobile SDK) ja alustan ohjelmistokehityspakettia (iOS tai Android), ja se suoritetaan mobiililaitteella. Mobiilisovellus yhdistetään DJI:n laitteeseen joko langattomasti käyttämällä Wi-Fi-verkkoa tai USB-kaapelilla. Valinta riippuu käytettävästä droneista. Ilma-aluksia varten mobiililaitte yhdistetään kauko-ohjaimen, joka yhdistyy langattomasti ilma-alukseen käyttämällä toista langatonta linkkiä. (11.) Kuva 8 havainnollistaa kahta eri tapaa, miten projektissa käytettävä Spark voidaan yhdistää mobiililaitteen kanssa.



Kuva 8. Yhteyden muodostaminen mobiililaitteen ja dronin välillä ohjaimen ollessa mukana ja ilman ohjainta (7).

Projektin toteutuksessa laitteiden yhdistämiseen käytettiin kuvassa näkyvää vasemmanpuoleista tapaa mukana olevan kauko-ohjaimen takia. Mobiililaitte yhdistetään kauko-ohjaimen Wi-Fi-verkkoon ja ohjain muodostaa yhteyden droniin käyttämällä toista langatonta yhteyttä.

Järjestelmät, anturit ja kompassi

Droonin ohjaamiseen ja hallintaan on käytössä useita antureita ja järjestelmiä, kuten IMU-järjestelmä (Inertial Measurement Unit), visiointijärjestelmä, kompassi ja RTK-paikannusjärjestelmä (Real Time Kinematic). Laitteen korkeuden seurantaan käytetään barometriä (10). IMU-järjestelmä sisältää kiihtyvyyssanturin ja gyroskoopin, joiden avulla mitataan kulmanopeutta ja lineaarista kiihtyvyyttä. IMU on herkkä, lämpötilasta riippuvainen järjestelmä, joka vaatii ajoittaista kalibrointia. Kalibroinnin voi aloittaa tarpeen mukaan DJI:n mobiiliohjelmistokehityspakkauksen API-toimintojen avulla. (12.) Järjestelmän voi kalibroida myös käyttämällä DJI GO 4 -nimistä sovellusta, joka on ladattavissa esimerkiksi Android-laitteelle Google Play -kaupasta.

Aluksen kompassi mittaa magneettisen kentän suuntaa, ja sitä käytetään laitteen suunnan määrittämiseksi suhteessa pohjoiseen. Droonia joudutaan joskus kalibroimaan, jos lennetään alueella, missä on magneettista häiriötä. Kompassin kalibrointiin sisältyy droonin kiertämistä pysty- ja vaakasuoraan atsimuutin kautta. (12.) Kompassin kalibroinnin voi myös suorittaa edellä mainituilla tavoilla. DJI:n tuotteet sisältävät lisäksi sisäänrakennetut kuluttajan satelliittipaikannusjärjestelmät, jotka käyttävät GPS- ja GLONASS-satelliittikonstellaatiota (Global Navigation Satellite System). Kuluttajatasen satelliittipaikannuksessa voi esiintyä useiden metrien paikannusvirheitä. (12.)

Esteiden havaitsemiseen ja paikan sekä nopeuden tarkkaan määrittämiseen voidaan käyttää droonin kameroita. Nämä kamerat yleensä sijaitsevat laitteen pohjassa paikannusta varten ja edessä esteiden havaitsemista varten, ja ne ovat erillisiä pääkamerasta, jota käytetään kuvien ja videoiden ottamiseen. Esteiden väistämiseen käytetään visiointijärjestelmää, joka parantaa lentämiseen liittyvää turvallisuutta ja alentaa todennäköisyyttä, että laitteella törmätään esteisiin (Projektissa käytetyssä DJI Sparkissa on edessä kamera, jonka avulla esteitä pystytään havaitsemaan laitteistodokumentaation mukaan 0,2–5 metrin etäisyydeltä). Järjestelmän rajoitukset tulee kuitenkin huomioida turvallisen lentämisen mahdollistamiseksi. Vaikeasti havaittavia kohteita ovat esimerkiksi pienet, kapeat ja ulkonäöltään liian tavalliset esteet. Alas osoittava kamera auttaa määrittämään suhteellista nopeutta ja paikkaa kuluttajakäytössä olevaa satelliittipaikannusjärjestelmää tarkemmin, ja sitä voidaan myös käyttää ympäristöissä, missä GPS ei ole saatavilla, kuten rakennusten sisällä (12).

2.4 Lennättämiseen liittyvät määräykset Suomessa ja Euroopassa

Suomessa on 1.1.2017 voimaanastunut liikenteen turvallisuusvirasto Trafín ilmailumääräys, joka koskee kauko-ohjatun ilma-aluksen ja lennokin lennättämistä. Määräystä sovelletaan yli 250 g painavien lennokkien ja kauko-ohjattujen ilma-alusten lennättämiseen lukuun ottamatta sotilasilmailua tai sisätiloissa lennättämistä. (13, s. 1.) Laitteen käyttötarkoitus määrittelee, luokitellaanko se lennokiksi vai kauko-ohjatuksi ilma-alukseksi, eli se voi olla kumpi tahansa, mikä taas riippuu jokaisen lennätyksen tarkoituksesta. Lennokiksi luokitellaan laite, jota käytetään harraste- tai urheilutoimintaan ilman mukana olevaa ohjaajaa. Kauko-ohjatuksi ilma-alukseksi luokitellaan laite, joka lentää ilman mukana olevaa ohjaajaa ja jota käytetään lentotyössä. Mikä tahansa muu harraste- ja urheilukäytön ulkopuolella oleva käyttö määritellään lentotyöksi. (14.)

Kauko-ohjatun ilma-aluksen ja lennokin lennättämislle on määräyksessä omat osionsa, joista tässä raportissa perehdytään jälkimmäiseen, koska projektissa käytettävän droonin lentoonlähtömassa on alle 3 kg ja laitteen toiminnallisuudet on kehitetty harrastetoimintana. Määräykset on määritelty seuraavasti:

- Lennätykset tulee suorittaa siten, että ulkopuolisille ihmisille ja heidän omaisuudelleen aiheutuva vaara on mahdollisimman pieni (13, s. 6).
- Ei ole sallittua lennättää lennokkia ulos kerääntyneen väkijoukon yläpuolella (13, s. 6).
- Tiheästi asutulla asutuskeskuksen alueella enintään 3 kg lentoonlähtömassaltaan painavan lennokin lennättäminen on sallittua, kun lennättäjä on varmistunut laitteen teknisestä kunnosta, perehtynyt alueeseen ja arvioinut lennättämisen olevan turvallista. Lennättäminen ei ole sallittua edellä mainitulla alueella, jos lennokin lentoonlähtömassa ylittää 3 kg. (13, s. 6.)
- Lennätykset eivät saa haitata, estää eikä vaarantaa paikalle saapuvan yksikön tai viranomaisen toimintaa hätä-, pelastus-, onnettomuus- tai vastaavassa poikkeustilanteessa (13, s. 6).
- Käyttäjän nimen ja yhteystietojen on käytävä ilmi lennokista (13, s. 7).

- Lennokin lennättäminen perustuu suoraan näköyhteyteen lennokin ja lennättäjän välillä. Ohjattavuus on säilytettävä koko ajan, ja lennätyksen on tapahduttava riittävän lähellä lennättäjää vallitseva sää sekä valoisuus huomioiden niin, että kyetään havaitsemaan muu ilmaliikenne ja arvioimaan suoran näköyhteyden perusteella luotettavasti väistämistarve ilman apuvälineitä. (13, s. 7.)
- Videolinkin avulla (FPV) lennättäminen on sallittua niin, että toiminnan turvallisuuden varmistamiseksi käytetään vähintään yhtä avustajaa, joka arvioi väistämistarpeen ja havainnoi esteet ja muun liikenteen luotettavasti ilman apuvälineitä. Havainnoitsijalla on oltava suora näköyhteys lennokkiin ja lennättäjään suora puheyhteys ilman viestintävälineitä. (13, s. 7.)
- Lennätyskorkeuden on oltava maan tai veden pinnasta alle 150 metriä lukuun ottamatta ilmailutiedotusjärjestelmässä julkaistuja, erikseen määritettyjä lennokkien lennätyspaikkoja (13, s. 7.)
- Lennokin lennättäminen lentoaseman läheisyydessä eli radiovyöhykkeellä (RMZ, Radio Mandatory Zone), lentopaikan lentotiedotusvyöhykkeellä (FIZ, Flight Information Zone) tai lähialueella (CTR, Control Zone) on sallittua maksimissaan 50 metrin korkeudella veden tai maan pinnasta, kun vaakasuora etäisyys on vähintään 5 kilometriä kiitotiestä. Mikäli lähempänä kiitotietä on tarve lennättää tai edellä mainituilla alueilla 50 metriä korkeammalla, on ilmaliikennepalvelun tarjoajan kanssa sovittava lennätyksistä erikseen. Lennättämisestä Utin (EFUT) ja Jyväskylän (EFJY) lentoasemien lähialueella on kaikissa tapauksissa kuitenkin sovittava ilmaliikennepalvelun tarjoajan kanssa erikseen. Lennättämisestä EFHK CTR SOUTH alapuolisella alueella on lisäksi sovittava Helsinki-Vantaan (EFHK) ilmaliikennepalvelun tarjoajan kanssa kaikissa tapauksissa. (13, s. 7.)
- Lennokin lennättämisessä on kiinnitettävä erityistä huomiota helikopterilentopaikkojen ja valvomattomien lentopaikkojen läheisyydessä tapahtuvaan ilmaliikenteeseen, ja paikallisia ohjeita on noudatettava soveltuvin osin (13, s. 7.)
- Lennokin on väistettävä kaikkia ilma-aluksia (13, s. 7.)

Suomen ilmailumääräysten lisäksi on tulossa voimaan myös muutaman vuoden kuluttua EU:n omat lennokkimääräykset. Vaikuttavina tekijöinä määräysten kehityksessä ovat olleet seuraavat asiat: multikoptereiden määrän lisääntyminen viimeisten vuosien aikana ja samalla kasvun seurauksena maassa ja ilmassa vaaratilanteiden lisääntyminen on johtanut siihen, että Euroopan ilmaturvallisuusviranomainen EASA on Euroopan komission pyynnöstä käynnistänyt hankkeen EU-tasoisien lennokkimääräyksen luomiseksi. Kaupalliset syyt ovat myös osa hankkeen vireille laittamista. Operaattorit ja laitteiden valmistajat halusivat aluksille yhtä, sopusoinnussa olevaa markkina-aluetta, joka olisi peruseriaatteiden mukainen EU:n alueella. (15.)

Aikataulutus menee seuraavasti: toukokuussa 2017 julkaistiin lopullinen määräysluonnos, ja uusi lennokkimääräys astuu voimaan viimeisimpien arvioiden mukaan kevään 2018 aikana. Nykymääräyksille on siinä annettu kolmen vuoden siirtymäaika. Kaudella 2022 noudatetaan siis ensimmäisen kerran uusia määräyksiä, ja se koskettaa kaikkia jäsenmaita, koska se on EU-asetus. Kaikki kansalliset lainsäädännöt, jotka ovat ristiriidassa asetuksen kanssa, tullaan korvaamaan sen mukaiseksi, ja voimaan se astuu heti. Valmistajat ja myyjät tekevät lennokit määräysten mukaisiksi. (15.)

Aluksiin liittyvät rajoitukset, toiminnot ja velvollisuudet ovat lyhyesti selitettynä ja nähtävissä esitteestä ”Leaflet”, joka tulee laitteen mukana. Lennättäjän tehtävänä on rekisteröidä itsensä ja lennokkinsa sekä tarpeen vaatiessa suorittaa verkkokurssi ja kurssiin liittyvä tutkinto. Nämä vaatimukset pätevät valtaosaan niistä, jotka hankkivat tulevaisuudessa aluksensa kaupasta ja keskittyvät valokuvaamiseen, videoiden tekemiseen ja muuhun vastaavaan toimintaan. (15.)

Liikenteen turvallisuusviraston ilmailumääräysten puitteissa prototyyppi on mahdollista kehittää. Drooni tulee olemaan lennättäjän jatkuvassa näköpiirissä, lentokorkeus on ohjelmoitavissa, alus on ohjaimella hallittavissa ja sitä ei lennätetä asutuskeskuksen tiheästi asutulla alueella eikä väkijoukon yläpuolella. Lennätysetaisyydet prototyypin toteutuksessa ovat suhteellisen lyhyet (joitain kymmeniä metrejä), ja droonia ei käytetä lentokenttien jne. välittömässä läheisyydessä. Lisäksi prototyyppi kehitetään harrastepohjalta, joten erillistä laitteen rekisteröintiä ei tarvita. EU:n määräyksiä ei toistaiseksi tarvitse huomioida.

3 Droonin käyttöönotto

Ennen droonin varsinaisten ohjelmallisten toimintojen käyttöönottoa perehdyttiin laitteen lento-ominaisuuksiin ja tutustuttiin, kuinka käytössä olevat järjestelmät toimivat lennon aikana. Kun yleiskuva ominaisuuksista saatiin muodostettua, aloitettiin ohjelmallisten toimintojen kehitys, joista ensimmäisenä keskityttiin mobiiliohjelmointikehityspaketin rekisteröintiin. Droonin ohjelmallisten komentojen käyttäminen edellyttää käytössä olevan mobiililaitteen onnistunutta rekisteröintiä. Lisäksi selvitettiin, kuinka mobiililaitteen ja droonin välille muodostetaan yhteys, jotta lennättämiseen tarkoitettuja toimintoja voitaisiin testata.

Henkilön kuvaamista varten myös droonin kameran toiminnot tuli toteuttaa ja kameran välittämän reaaliaikaisen kuvan tuli olla nähtävissä mobiililaitteella. Tiedonsiirtoprotokollana osapuolten välisessä kommunikaatiossa käytettiin MQTT:tä (Message Queue Transport Telemetry). Ohjelmallisten toimintojen kehitysympäristönä projektin toteutuksessa toimi Android Studio, ja komentoja suoritettiin Android-käyttöjärjestelmällä varustetun mobiililaitteen kautta.

3.1 Lennätyksen ja järjestelmien alustava testaus

Lento-ominaisuuksien ja valvovien järjestelmien testaamista varten mobiililaitteelle ladataan DJI GO 4 -sovellus. Sovelluksen avulla lennättäminen tapahtuu FPV-näkökulmasta (First Person View), eli käyttäjä näkee puhelimessaan reaaliaikaista kuvaa droonin näkökulmasta katsottuna. Myös kameran toimintoja, kuten valokuvaamista ja videoidamista, pystyttiin testaamaan samassa yhteydessä. Sovelluksen avulla on lisäksi mahdollista käyttää valmiita DJI:n kehittämiä älykkäitä lentotoimintoja, joihin kuuluu mm. kohteen aktiiviseuranta (ActiveTrack), käden liikkeiden mukaan lennättäminen (Gesture) ja kosketa näyttöä ja lennä -toiminto (TapFly).

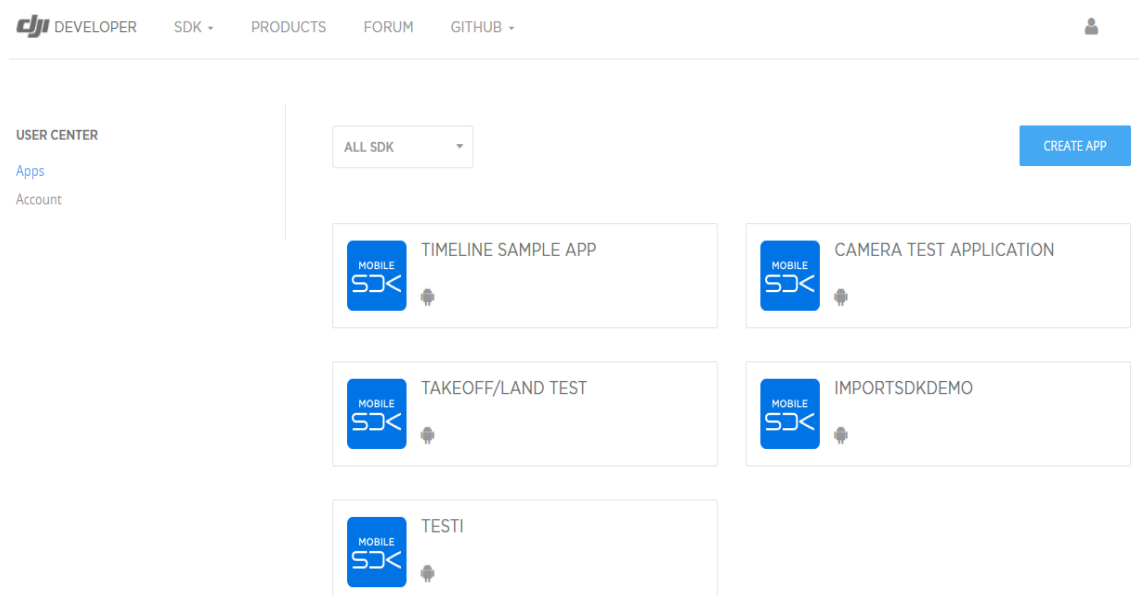
Älykkäitä lentotoimintoja on mahdollista käyttää myös ohjelmistokehityspaketin API-rapaintaa hyödyntämällä omassa mobiilisovelluksessa. Projektissa toimintoja ei voitu kuitenkaan käyttää, koska ne edellyttävät käyttäjältä vuorovaikutusta mobiililaitteen kanssa. Prototyypin tarkoituksena on kehittää itsehallinnollinen järjestelmä, jossa käyttäjän ei tarvitse olla sovelluksen kanssa tekemisissä, vaan kaikki lentämiseen liittyvät toiminnot

ovat automatisoituja. Käyttäjällä tarvitsee olla ainoastaan mahdollisuus ottaa droni manuaalisesti takaisin hallintaansa häiriötilan tapahduttua. Häiriö voi liittyä esimerkiksi droonin arvaamattomaan käyttäytymiseen lennon aikana.

Myös visiointijärjestelmän toimintaan tutustuttiin lennätyksen yhteydessä. Järjestelmän toiminta testattiin tavalla, jossa droni lennätettiin nokka edellä lähelle seinää. Esteen ilmaannuttua lento keskeytyi. Tätä testiä kokeiltiin seinän lisäksi pienemmällä kohteella. Kohteeksi valittiin projektin toinen kehittäjä, ja seurauksena oli sama lopputulos kuin aikaisemminkin, eli lentoa ei pystytty jatkamaan, ennen kuin kohde siirtyi pois edestä tai este kierretään.

3.2 Droonin toiminnallisuuksien avaaminen mobiiliympäristöön

Droonin perustoimintoihin perehtymisen jälkeen kehitettiin ensimmäinen Android-sovellus, jonka avulla testattiin laitteen ohjelmallisten toimintojen avaamista mobiililaitteelle. Toimintojen avaaminen koostuu useasta eri vaiheesta, jotka menevät seuraavasti: Käyttäjän tulee rekisteröityä DJI:n sivustolla kehittäjäksi. Rekisteröitymisen jälkeen DJI:n käyttäjäkeskuksessa luodaan sovellus, joka yhdistetään mobiililaitteelle kehitettävän sovelluksen kanssa. Kuva 9 havainnollistaa käyttäjäkeskukseen luotuja sovelluksia, joista jokainen liittyy aina yhteen tiettyyn Android-sovellukseen.



Kuva 9. DJI:n käyttäjäkeskukseen luotuja sovelluksia.

Sovelluksen luomisen yhteydessä käyttäjää pyydetään täyttämään sovellukseen liittyviä tietoja siitä, missä DJI:n API-rajapinnan tarjoamia toimintoja halutaan hyödyntää. Kuva 10 havainnollistaa sovelluksen luomisen yhteydessä täytettäviä tietoja, joista oleellisimmat ovat ohjelmistokehityspaketin tyyppi (Mobile SDK tai Onboard SDK), ohjelmistoalusta (Android tai iOS) ja paketin nimi, joka on esimerkiksi Android-ympäristössä nähtävissä sovelluksen AndroidManifest.xml-tiedostossa.

CREATE APP

SDK	Mobile SDK
APP Name	
Software Platform	Android
Package Name ⓘ	
Category	
Description	

CANCEL CREATE

Kuva 10. DJI-sovelluksen luominen.

Sovelluksen luomisen yhteydessä käyttäjää pyydetään vielä varmentamaan tapahtuma sähköpostitse. Varmentamisen jälkeen sovelluksesta on saatavilla sovellusavain, joka mahdollistaa rekisteröimisen mobiiliympäristössä. Kuva 11 havainnollistaa luotua DJI-sovellusta ja siitä saatavaa sovellusavainta.

USER CENTER
Apps
Account

< Back

APP INFORMATION

SDK Type	Mobile SDK
APP Name	Timeline sample app
Software Platform	Android
Bundle Identifier	com.dji.sdk.sample
App Key	[REDACTED]
Category	Automated flight testing
Description	Sample code for testing timeline mission.

EDIT DELETE

Kuva 11. Luodun sovelluksen tiedot.

Sovellusavain lisätään Android-sovelluksen AndroidManifest.xml-tiedostoon. Kuva 12 havainnollistaa sovellusavaimelle tarkoitettua kohtaa. Myös muut kuvassa näkyvät elementit tulee lisätä samaan tiedostoon.

```

<!-- DJI SDK -->
<uses-library android:name="com.android.future.usb.accessory" />
<meta-data
    android:name="com.dji.sdk.API_KEY"
    android:value="Please enter your App Key here." />
<activity
    android:name="dji.sdk.sdkmanager.DJIAoaControllerActivity"
    android:theme="@android:style/Theme.Translucent" >
    <intent-filter>
        <action android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED" />
    </intent-filter>
    <meta-data
        android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
        android:resource="@xml/accessory_filter" />
</activity>
<service android:name="dji.sdk.sdkmanager.DJIGlobalService" >
</service>
<!-- DJI SDK -->

```

Kuva 12. Sovellusavaimen lisääminen AndroidManifest.xml-tiedostoon (16).

Mobiiliohjelmistokehityspaketin rekisteröintiä varten tarvitaan myös mobiililaitteelta melko suuri määrä eri käyttöoikeuksia. Sovellukselle on sallittava käyttöoikeudet, jotta DJI:n ohjelmistokehityspaketti voi toimia (16). Android-sovelluksen käynnistyessä käyttöoikeudet tarkistetaan, ja mikäli käyttäjällä on riittävät oikeudet, voidaan mobiiliohjelmistokehityspaketin rekisteröinti aloittaa. Kuvassa 13 ovat nähtävissä kaikki oikeudet, jotka ohjelmistokehityspaketti tarvitsee toimiakseen mobiililaitteella.

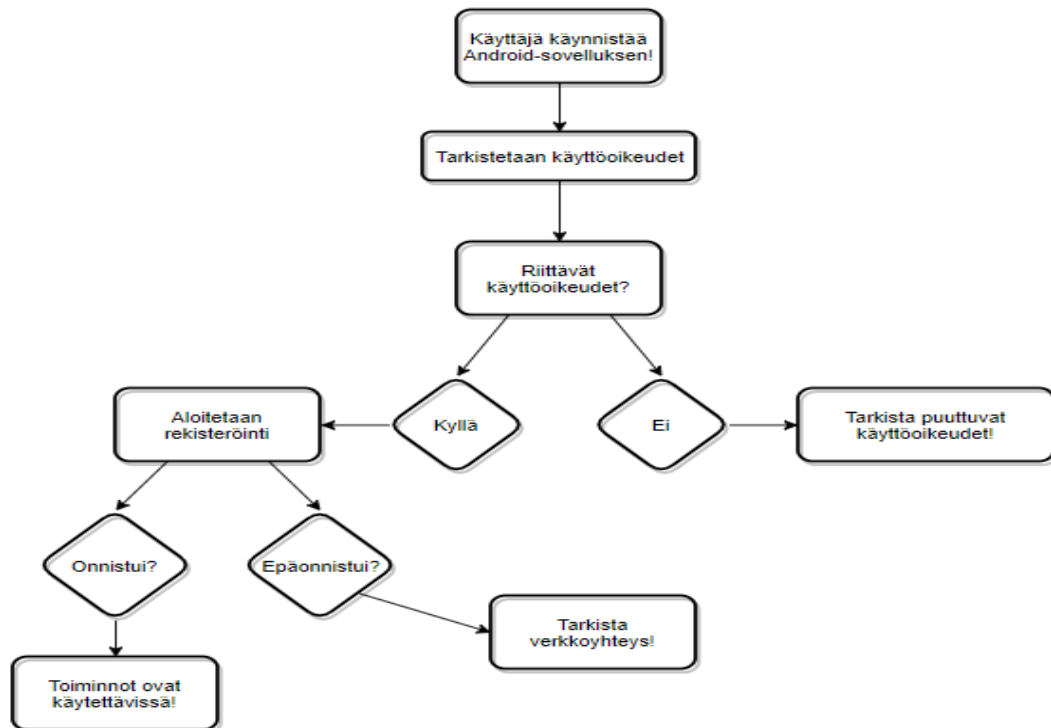
```

<!-- Permissions and features -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />

```

Kuva 13. Tarvitavat käyttöoikeudet rekisteröintiä varten (16).

Android-ohjelmassa lisätään riippuvaisuudet build.gradle-tiedostoon (Module:app), jotta viimeisimmät DJI:n Android-ohjelmistokehityspaketit saadaan tuotua sovelluksen käytettäväksi. Lisäksi pääaktiviteetti-luokan (MainActivity) tarvitsee rekisteröidä sovellus saadakseen valtuutukset käyttää mobiiliohjelmistokehityspakettia. Ohjelmistokehityspaketin odottamat takaisinkutsu-metodit (callback) tulee myös implementoida. (16.) Kuva 14 havainnollistaa rekisteröintiprosessia Android-sovelluksessa.



Kuva 14. Ohjelmistokehityspaketin rekisteröinnin kulku Android-sovelluksessa.

Rekisteröinnin onnistuminen Android-ympäristössä edellyttää siis seuraavia asioita tiivistettynä: DJI:n käyttäjäkeskukseen on luotu onnistuneesti sovellus ja saatu sovellusavain on asetettu Android-sovelluksen AndroidManifest.xml-tiedostossa sille tarkoitettuun paikkaan. Android-sovelluksessa on lisätty build.gradle-tiedostoon tarvittavat riippuvaisuudet viimeisimmän DJI-ohjelmistokehityspaketin tuomiseksi sovelluksen käyttöön. Sovellukselle on annettu riittävät käyttöoikeudet rekisteröinnin mahdollistamiseksi ja ohjelmistokehityspaketin odottamat takaisinkutsu-metodit on implementoitu ohjelmaan oikein. Kuva 15 havainnollistaa esimerkkiä onnistuneesta DJI:n ohjelmistokehityspaketin rekisteröinnistä Android-laitteella.



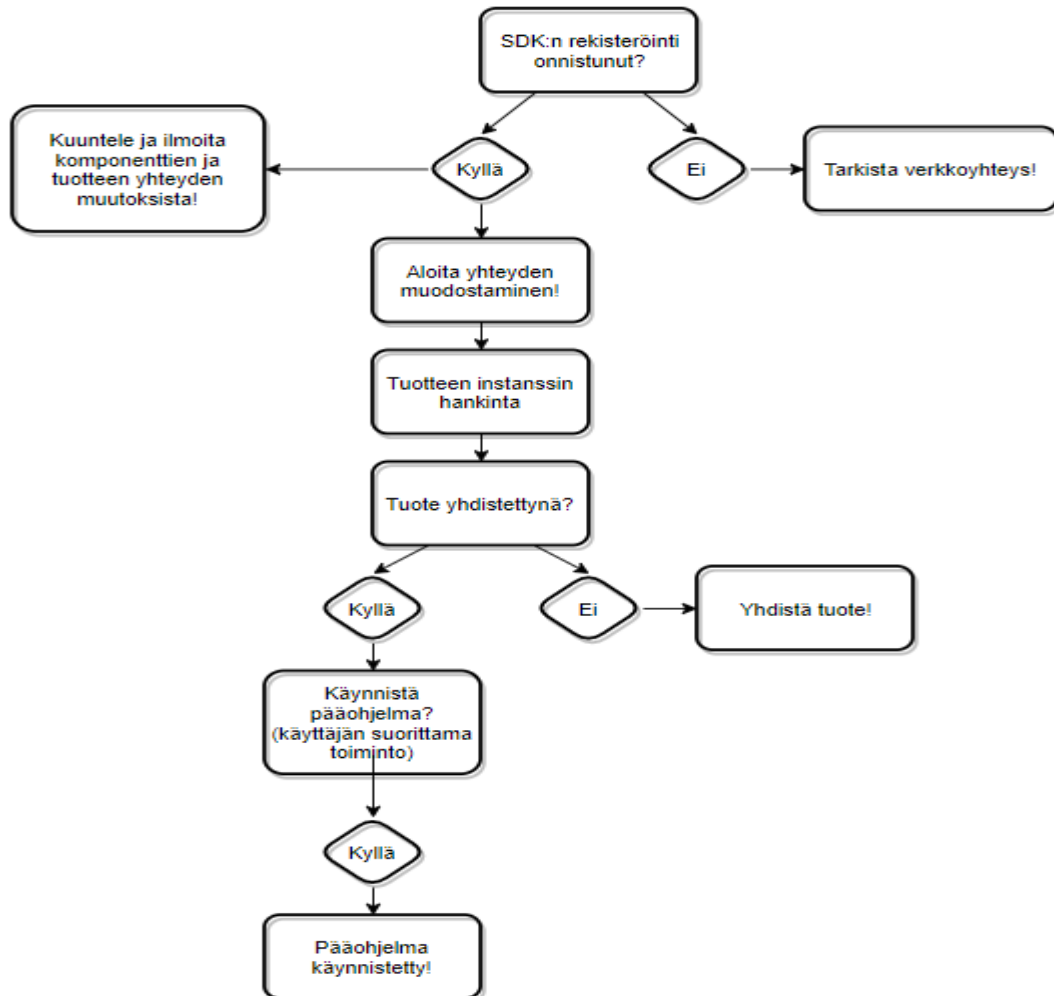
Kuva 15. Ohjelmistokehityspaketin onnistunut rekisteröinti Android-laitteella (16).

DJI:n kehittäjä sivustolta löytyy mobiiliohjelmistokehityspaketin käyttöönottoon liittyviä esimerkkiprojekteja, joissa käyttöönottoon sisältyvät vaiheet on selitetty molemmille, Android- ja iOS-alustoille. Esimerkkejä hyödynnettiin myös prototyypin kehityksessä. Toimituksen aikana kuitenkin ilmaantui rekisteröintiin liittyviä ongelmia, eikä rekisteröintiä saatu suoritettua. Ongelman syyksi selvisi liian vanhalla Android-versiolla varustettu mobiililaitte (versio 4.4.2). Rekisteröinnin onnistuminen edellyttää minimissään versiolla 5.0.0 varustettua laitetta. Luotu ohjelma asennettiin projektin toisen kehittäjän uudempaan matkapuhelimeen, ja siinä rekisteröinti onnistui.

3.3 Mobiilisovelluksen ja dronin yhdistäminen

Ennen kuin ohjelmoitavia, lennättämiseen liittyviä toimintoja pystyttiin testaamaan, oli perehdyttävä siihen, kuinka dronin ja mobiililaitteen välille muodostetaan yhteys. Tar-

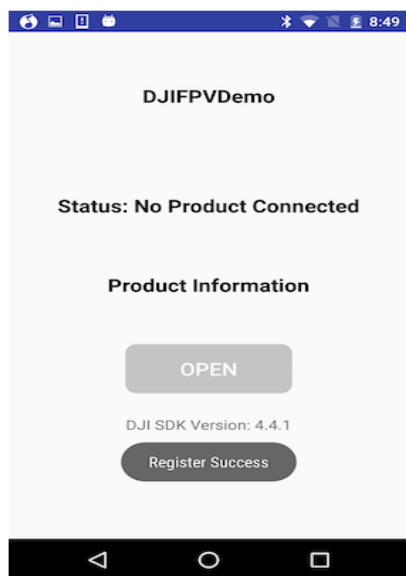
vittavia toimintoja yhteyden muodostamiseksi tutkittiin DJI:n sivustolta löytyvien esimerkiprojektien pohjalta ja API-rajapinnan dokumentaation avulla. Toimintojen käyttöönotto sujui suoraviivaisesti ja ilman ongelmia käytettävissä olevan materiaalin ansiosta. Kuva 16 havainnollistaa yhteyteen liittyvien toimintojen suoritusta Android-sovelluksessa. Yhteyden muodostaminen laitteiden välillä aloitetaan vasta, kun mobiiliohjelmistokehityspaketin rekisteröinti on suoritettu onnistuneesti.



Kuva 16. Yhteyden muodostaminen Android-sovelluksen ja dronin välillä.

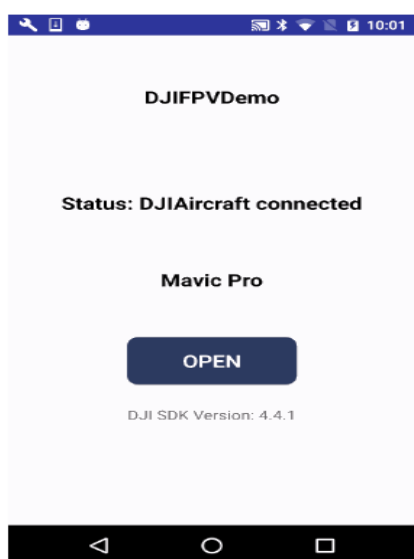
Dronin yhdistämistä varten mobiililaitteelle luotiin ulkoasu, josta on nähtävissä ohjelmistopakettien rekisteröinnin onnistuminen tai epäonnistuminen, käytössä olevan paketin versio, yhdistettävä laite (Spark, Mavic, Phantom jne.) ja yhteyden tila. Ulkoasu toimii myös aloitussivuna Android-sovelluksen käynnistyttyä. Kuva 17 havainnollistaa aloitus-sivua, kun ulkopuolisia laitteita ei ole vielä yhdistetty sovelluksen kanssa. Ohjelman käynnistyessä ohjelmistokehityspaketin rekisteröinti suoritetaan, ja sen jälkeen jäädään

odottamaan, että käyttäjä yhdistää mobiililaitteensa käytössä olevaan kauko-ohjaimen, jonka puolestaan pitää olla yhdistettynä dronin kanssa.



Kuva 17. Onnistunut rekisteröinti aloitussivulla (17).

Kun mobiililaitte on yhdistetty kauko-ohjaimen Wi-Fi-verkkoon, yhdistämiseen liittyvät tiedot (laite, tila) päivitetään aloitussivulla. Myös avauspainikkeen tila päivitetään sivulla harmaasta siniseksi indikoimaan, että käyttäjä voi aloittaa pääohjelman suorittamisen. Kuva 18 havainnollistaa aloitussivua, kun yhteys on muodostettu oikein ja on aktiivisessa tilassa mobiililaitteen ja dronin välillä.

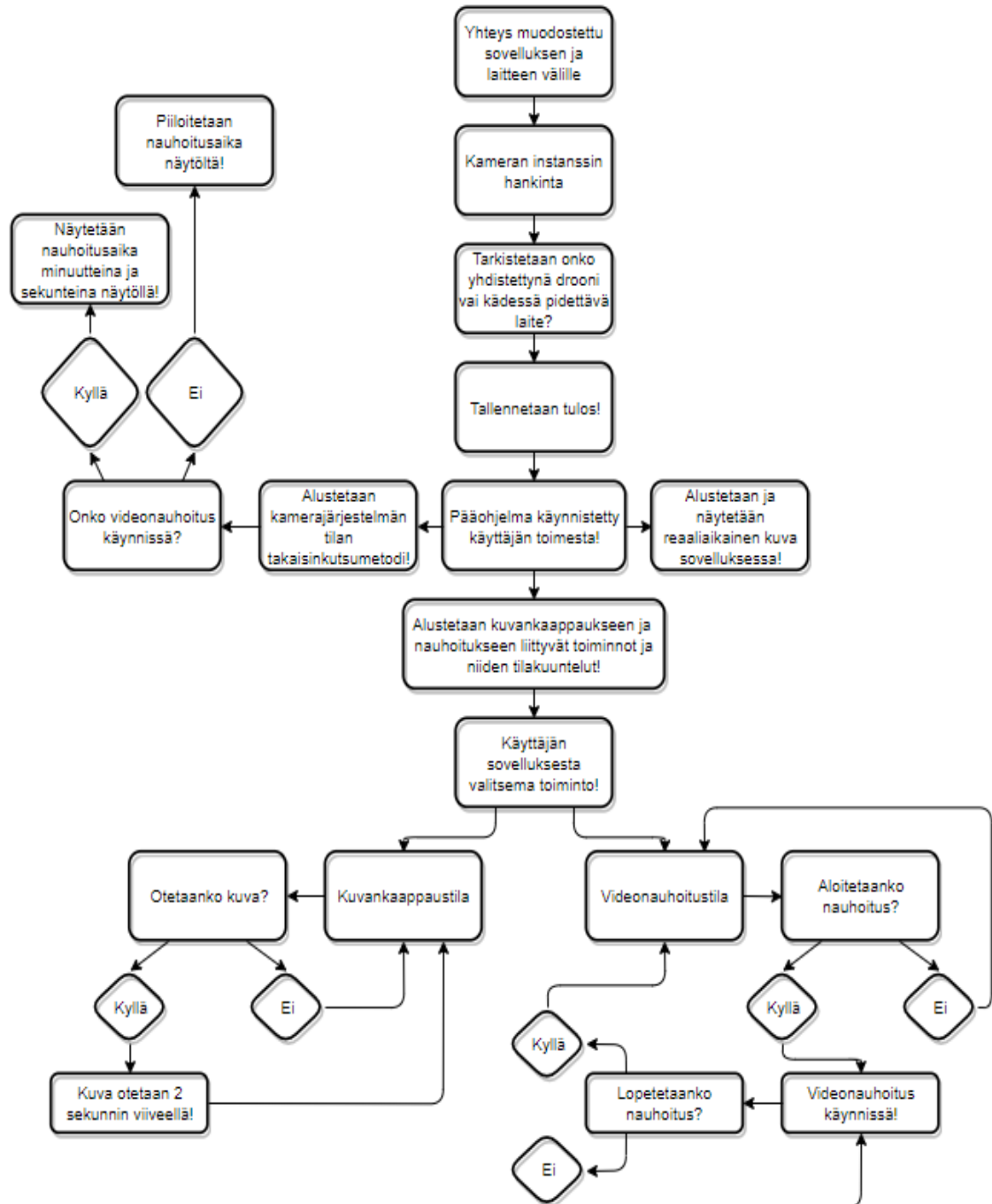


Kuva 18. Yhteys muodostettu dronin ja sovelluksen välille (17).

Yhteyden muodostamiseen tarvittavien toimintojen luominen oli projektin muihin osuuksiin nähden helpointa saatavilla olevan materiaalin ja esimerkkien ansiosta. Lisäksi kehittämistä helpotti se, että mobiililaitteen versioihin liittyvät ongelmat oli saatu selvitettyä jo ohjelmistokehityspaketin rekisteröinnin yhteydessä. Onnistuneen yhteyden muodostamisen seurauksena oli mahdollista aloittaa varsinaisten päätoimintojen kehittäminen ja testaaminen, joista ensimmäisenä perehdyttiin kameraan.

3.4 Kameran käyttöönotto

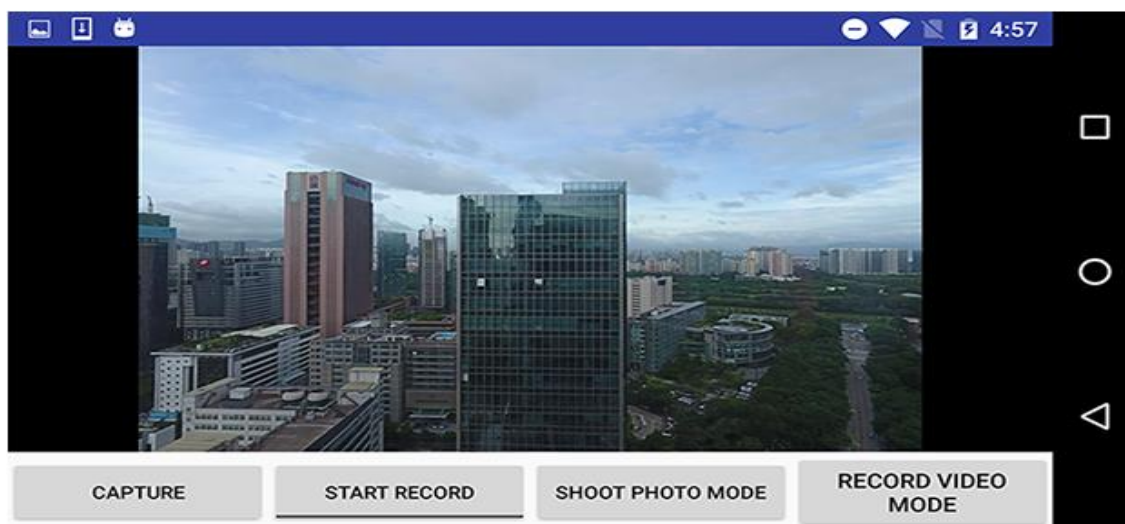
Käytössä oli 1920*1080 -videoresoluutiolla varustettu 12-megapikselinen droonin kamera, jota käytetään projektissa maastossa turvarajan ylittäneen henkilön kuvaamiseen. Käyttäjä näkee droonin lähettämää reaaliaikaista kuvaa mobiilisovelluksestaan ja voi myös halutessaan tallentaa lennon aikana näkemäänsä kahdella eri tavalla, joko video-kuvaamalla tai ottamalla kuvakaappauksia. Tallennettua kuvaa voitaisiin hyödyntää myöhemmin esimerkiksi henkilön terveydentilan analysoinnissa, maaston kartoituksessa jne. Kuva 19 havainnollistaa kameran toimintoihin liittyvää käyttöönottoa Android-sovelluksessa. Kuten aikaisemmin, myös kameran toteutuksen pohjatietona hyödynnettiin sekä esimerkkiprojekteja että API-rajapinnan dokumentaatiota.



Kuva 19. Kameraan liittyvät toiminnot Android-sovelluksessa.

Pääohjelman käynnistyttyä käyttäjän on mahdollista valita kahden eri toimintatilan väliltä, kuvakaappauksen tai videonauhoituksen. Lisäksi reaaliaikaista kuvaa välitetään jatkuvasti dronin kamerasta mobiililaitteen näytölle. Eri kuvaamistilat ovat valittavissa näytöllä olevista painikkeista, ja esimerkiksi videokuvaamistilanteessa nauhoituksen käynn-

nistyttyä näytöllä näytetään myös tallennuksen kesto. Kuvankaappauksessa kuvan ottamista viivästetään, koska kamera tarvitsee aikaa valokuvatilan asettamiseen (17). Nämä kuvaamiseen liittyvät toimintatilat ovat prototyypissä lisäominaisuuksia, joiden avulla kameralin toimintoja voidaan havainnollistaa. Kuva 20 esittelee kameralin näkymää ja valittavissa olevia kuvaamistiloja.



Kuva 20. Havainnollistava näkymä Android-sovelluksesta katsottuna (17).

Mobiilisovelluksesta nähtävää reaaliaikaista kuvaa oli myös tarkoitus striimata esimerkiksi YouTubeen, jotta käyttäjä pystyisi seuraamaan lennätyksen kulkua etäällä olevasta päätteestä. Toteutuksessa kuitenkin esiintyi seuraavanlainen ongelma: Spark ei tue reaaliaikaista striimaamista Android-laitteella, koska mobiilidataa ei voida käyttää samanaikaisesti silloin, kun Android on yhdistettynä Wi-Fi-verkkoon (18). Prototyypissä tämä tilanne esiintyy, kun käyttäjä yhdistää mobiililaitteensa kauko-ohjaimen Wi-Fi-verkkoon. Vaikka striimaamiseen käytettäisiin jotain muuta keinoa, ongelmana on, että pääsyä verkkoon ei ole, koska mobiilidataa ei voida käyttää samanaikaisesti.

Tilannetta koetettiin ratkaista asettamalla Android-puhelimen kehittäjäasetuksista ”Mobiilidata aina aktiivinen” -asetus päälle, mutta tällä ei ollut vaikutusta verkkoyhteyden toimintaan. Ajan säästämiseksi päätettiin siirtyä kehittämään muita projektin päätoimintoja, ja ongelman pariin palattaisiin vasta, kun muut lennättämisen kannalta oleelliset asiat oli saatu luotua.

3.5 Tiedonsiirto palvelimen ja mobiilisovelluksen välillä

Tiedonsiirtoon palvelimen ja mobiililaitteiden välillä käytettiin projektissa MQTT-protokollaa. MQTT on suosittu tiedonsiirto- tai viestintäprotokolla, ja sen käyttö on yleistynyt laitteiden välisessä kommunikaatiossa (machine-to-machine) tai tuttavallisemmin sanottuna esineiden internetissä (Internet of Things). MQTT on viestintäprotokolla, joka käyttää julkaisu- ja tilausmallia (publish & subscribe) ja on esineiden internet -arkkitehtuurin kannalta tärkeä, niin kutsuttu ”kevyt” viestintäprotokolla, jossa käytettävät laitteet ovat yleensä matalatehoisia antureita, mikrokontrollereita jne. (19.)

Eräs useista MQTT-protokollan käyttötarkoituksista on lähettää anturidataa sulautetuista järjestelmistä, ja joskus tätä dataa halutaan lähettää mobiililaitteelle, mikä voisi auttaa käyttäjiä tarkkailemaan tärkeitä asioita etäisyyden päästä (19). Projektissa kerätään paikannettavan henkilön koordinaatteja mobiililaitteen avulla ja kerätty tieto välitetään palvelimelle, joka näyttää henkilön sijainnin kartalla. Palvelin välittää myös henkilön koordinaatit toiselle mobiililaitteelle, joka ”herättää” dronin rajanylityksen tapahduttua ja lennättää sitä lähtöpaikan ja henkilön välillä. Lennätyksen aikana lisäksi dronin sijainti välitetään takaisin palvelimelle kartalla näyttämistä varten.

MQTT:n käyttöönotto edellyttää riippuvaisuuksien lisäämistä Android-sovellukseen eli Eclipsen tarjoaman Paho MQTT clientin ja Android-palvelun kirjastojen lisäämistä ohjelmaan. Virallisen Paho Eclipse Githubin mukaan kirjaston voi asentaa sovellukseen joko käyttämällä Maven Gradlea tai asentamalla sen suoraan lähteestä. (19.) Kuva 21 havainnollistaa asennuksessa käytettävää Maven Gradlea ja tarvittavia tietoja.

```

18  allprojects {
19      repositories {
20          google()
21          jcenter()
22          maven {
23              url "https://repo.eclipse.org/content/repositories/paho-snapshots/"
24          }
25          maven { url "https://jitpack.io" }
26      }
27  }

```

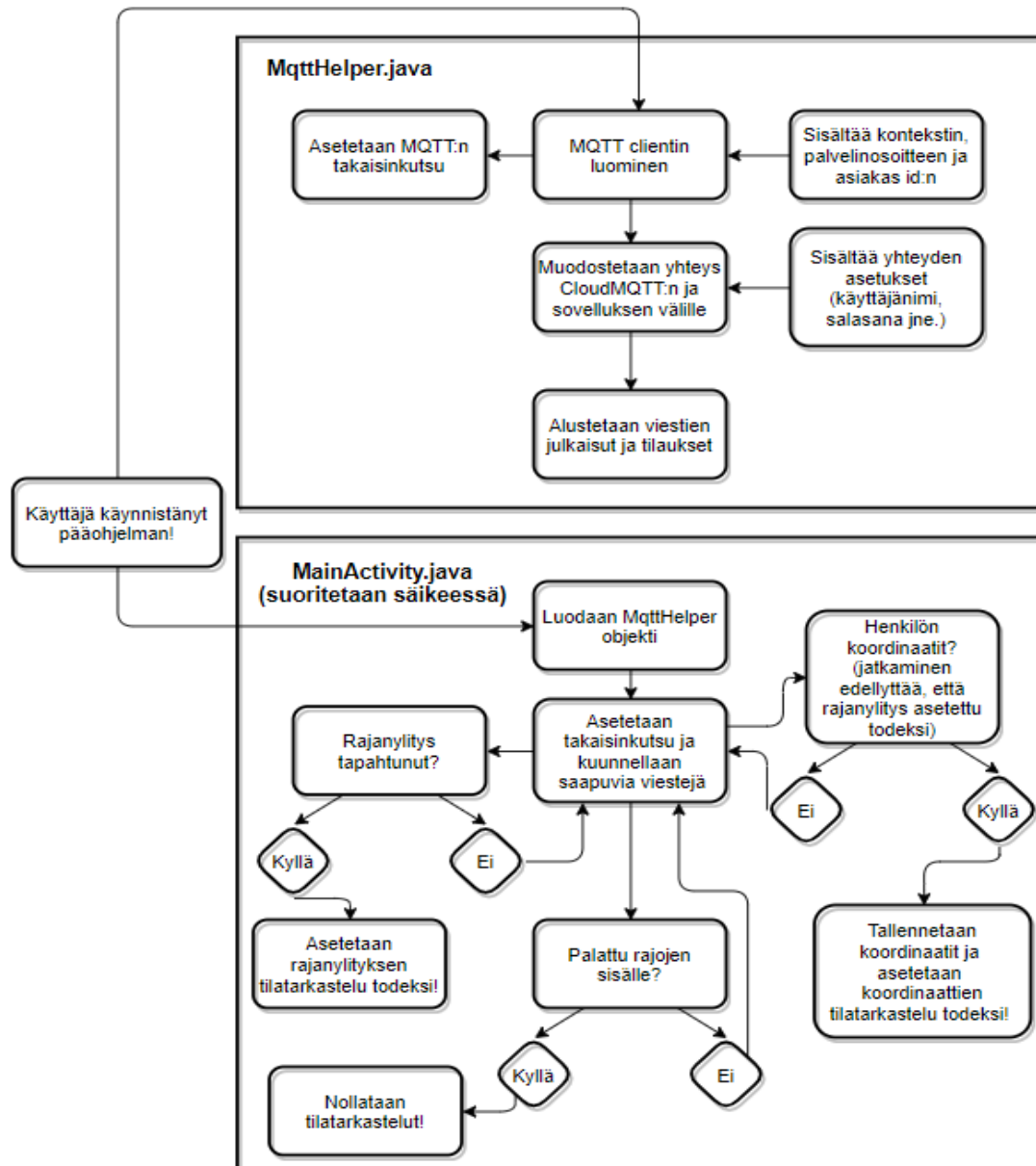
Kuva 21. Build.gradle(project)-tiedostoon sijoitettavat tiedot.

Kuva 22 havainnollistaa MQTT:n käyttöönottoon liittyviä, Android-sovellukseen liitettäviä riippuvaisuuksia.

```
33 dependencies {  
34     implementation fileTree(include: ['*.jar'], dir: 'libs')  
35     compile 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0'  
36     compile 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.1'
```

Kuva 22. Build.gradle(app)-tiedostoon sijoitettavat tiedot.

Tiedonsiirtoa varten tarvitaan MQTT-välittäjä (MQTT broker). On olemassa joitakin pilvipalveluita hyödyntäviä MQTT-välittäjiä, kuten HiveMQ ja CloudMQTT (19), joista projektin toteutuksessa käytettiin jälkimmäistä. Projektin toinen kehittäjä kertoo CloudMQTT:hen tehtävistä asiakasrekisteröinneistä ja muista välittäjän käyttöönottoon liittyvistä asioista insinööriyössään. Kuva 23 havainnollistaa MQTT-palvelun käyttöönottoa ja toimintatapaa Android-sovelluksessa.



Kuva 23. MQTT-tiedonsiirto-protokollan käyttö Android-sovelluksessa.

MQTT:n käyttöönotto menee siis seuraavasti: MqttHelper-luokassa alustetaan kaikki tiedonsiirtoa varten tarvittavat asiat, joihin kuuluvat esimerkiksi asiakas-id, käyttäjänimi, salasana ja palvelun muut toiminnot, kuten automaattisen uudelleen yhdistämisen määrittelyt jne. Lisäksi luodaan julkaisua ja tilaamista varten tarvittavat toiminnot. Pääohjelmassa saapuvia viestejä kuunnellaan, ja mikäli viesti sisältää tietoa turvarajan ylityksestä tai henkilön koordinaateista, toimitaan ohjelmaan asetettujen ehtojen mukaisesti.

MQTT:n toteutuksessa tiedonsiirron kanssa esiintyi sama ongelma, joka oli myös aikaisemmin mainitussa kameran kuvan striimaamisessa. Koska mobiililaitte yhdistetään

kauko-ohjaimen Wi-Fi-verkkoon, ei mobiilidata ole käytettävissä kyseisenä aikana, eli saapuvaa tietoa palvelimelta ei pystytä vastaanottamaan eikä droonin sijaintitietoja voida myöskään lähettää. Tämä oli haastava ongelma, sillä prototyypin toteutus perustuu useiden laitteiden väliseen kommunikatioon. Ratkaisua koetettiin selvittää tutkimalla, kuinka mobiilidatan ja Wi-Fi-verkon yhtäaikainen käyttö voitaisiin toteuttaa ohjelmallisesti Android Studiolla, mutta tiedonhaut eivät tuottaneet tulosta.

Ongelmaan löytyi kuitenkin vaihtoehtoinen ratkaisu. Kun tutkittiin valmiita sovelluksia, joissa mobiilidatan ja Wi-Fi-verkon yhtäaikainen käyttö oli jo toteutettu, vastaan tuli Speedify-niminen ohjelma. Speedifyn ainutlaatuinen kanavan sidos -teknologia (channel bonding technology) sallii käyttäjän käyttää turvautusti useita internetyhteyksiä samanaikaisesti kaistanleveyden lisäämiseksi, alhaisemman viiveen ja paremman luotettavuuden saavuttamiseksi (20). Kuva 24 havainnollistaa, miltä sovellus näyttää ohjelman käynnistyttyä.



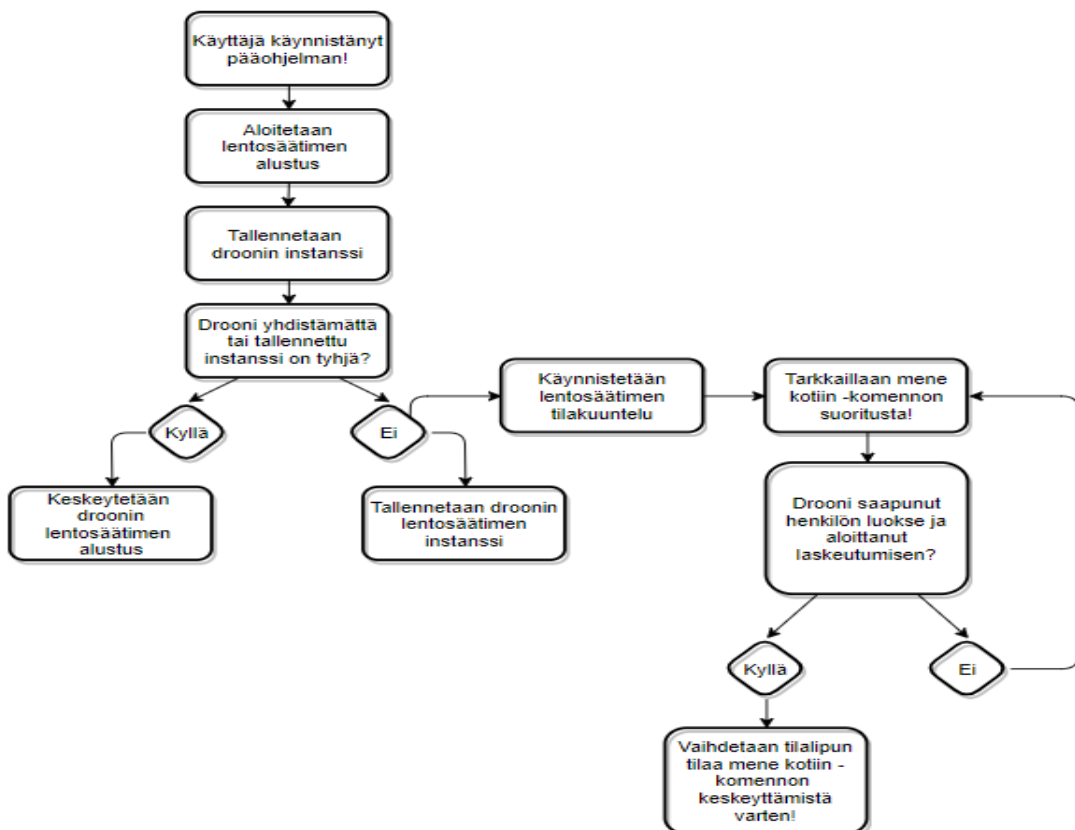
Kuva 24. Speedify (21).

Speedifyn tarjoamaa palvelua on mahdollista käyttää ilmaiseksi, jolloin käytössä on 1 GB dataa kuukaudessa. Kuukausittainen datamäärä oli riittävä prototyypin toteutuksessa, sillä ohjelmaa käytettiin pääasiassa ainoastaan silloin, kun droonin lennättämiseen liittyviä toimintoja testattiin. Speedifyn käytössä ilmeni testausvaiheiden aikana

myös satunnaisia yhteysongelmia. Sovellus ei ajoittain saanut yhdistettyä käytössään oleviin palvelimiin, ja toisinaan mobiililaitteen oma verkko ei toiminut kunnolla, mikä taas esti yhteyden muodostamisen Speedifyn palveluihin. Yhteysongelmia projektin toteutuksen aikana ilmeni kuitenkin suhteellisen vähän, ja valtaosan ajasta testauksia pystyttiin suorittamaan halutulla tavalla.

3.6 Lentämiseen liittyvät toiminnot

Onnistuneen rekisteröinnin ja yhteyden muodostamisen seurauksena käyttäjä pystyy hyödyntämään dronin toimintoihin liittyviä ominaisuuksia. DJI:n API-rajapinnan lentosäädin-luokan (FlightController) avulla esimerkiksi dronin lentoonlähtöä, laskeutumista, moottoreita, mene kotiin -toimintoa (GoHome), dronin lentokorkeutta ja useita muita ominaisuuksia pystytään hallinnoimaan. Toimintojen käyttäminen edellyttää lentosäätimen alustusta mobiilisovelluksessa. Myös komentojen suoritusta ja tilaa on mahdollista tarkkailla lentosäädintila-luokan (FlightControllerState) avulla. Kuva 25 havainnollistaa dronin lentosäätimen ja sen tilakuuntelun käyttöönottoa Android-sovelluksessa.



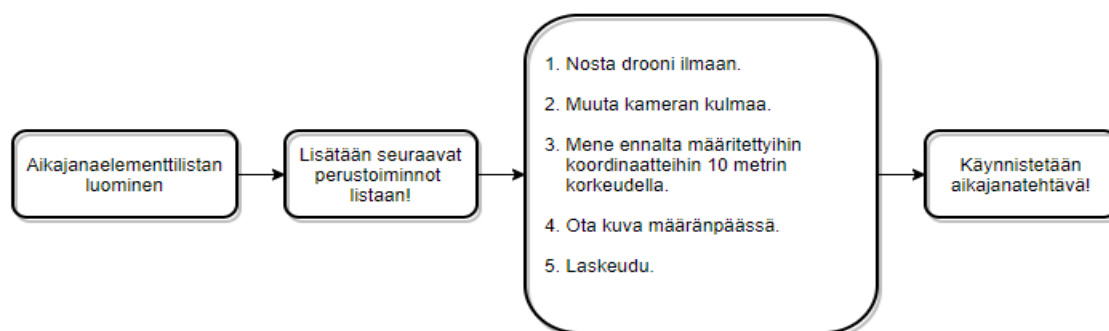
Kuva 25. Dronin lentosäätimen ja sen tilakuuntelun alustus.

Alustuksen onnistumisen jälkeen droonin komponenttien ohjaukseen liittyvät toiminnot ovat käyttäjän hallittavissa ja niiden käyttäminen on suhteellisen yksinkertaista, sillä ne ovat valmiita metodeja, joita käyttäjä kutsuu tarpeen mukaan sovelluksessaan. Lentosäätimen alustuksen yhteydessä kehitettiin myös ensimmäiset lennättämiseen liittyvät toiminnot, joihin sisältyi droonin lento-ohjauksen ja laskeutumisen. Android-sovellukseen kehitettiin painonapit, joiden avulla lento-ohjaukseen ja laskeutumiseen saatiin myös onnistuneesti testattua. Kun komentojen toimivuus saatiin todennettua, pystyttiin prototyypin kehitystä jatkaa kohti viimeistelyä versiota.

Lennättämisen alkuperäinen toteutustapa

Alkuperäiseen toteutustapaan sisältyi aikajana-ohjauksen hyödyntäminen ja aikajanaelementteihin kuuluvan mene määränpäähän -toiminnon (GoToAction) käyttäminen osana aikajana-ohjaukseen. Aikajanaan liittyvät toiminnot ovat osa tehtävähallinta-luokkaa (MissionControl). Tehtävähallinta hoitaa kaikki tehtävien suoritukset. Yksittäisiä tehtäviä voidaan suorittaa käyttämällä tehtäväoperaattoreita, tai useita tehtäviä ja toimintoja voidaan jaksottaa käyttämällä aikajanaa. Tehtävähallinnan aikajana koostuu aikajanaelementtisarjoista, jotka voivat olla kuuma piste -tehtäviä (Hot Point), reittipiste-tehtäviä (Waypoint) tai objekteja, jotka seuraavat aikajanaelementti-protokollaa. (22.)

Jos tehtävää suoritetaan aikajana-ohjauksella, tehtäväoperaattoria voidaan silti käyttää tehtävään liittyvän tilatiedon hakemiseen ja tehtävän suoritukseen liittyviä parametrejä voidaan myös muuttaa (22). Prototyypin kehityksessä aikajanaelementteistä koostettiin perustoimintojen pohjalta aikajana-ohjaukseen, joka suoritettiin Android-sovelluksessa elementti kerrollaan. Kuva 26 havainnollistaa sovellukseen luotua aikajana-ohjaukseen ja siinä käytettyjä perustoimintoja.



Kuva 26. Android-sovellukseen luotu aikajana-ohjaukseen.

Aikajanatehtävän suoritus alkoi lupaavalla tavalla. Drooni nosti itsensä ensin ilmaan, minkä jälkeen kameran kulma muuttui ennalta määritettyyn asteeseen. Kun vuoroon tuli koordinaatteihin siirtyminen halutulla korkeudella (toiminnon suorittamisessa käytettiin aikaisemmin mainittua mene määränpäähän -komentoa), ohjelman suoritus keskeytyi eikä jälkimmäisiäkään toimintoja voitu suorittaa. Ohjelman keskeytymisen syyksi epäiltiin ensin droonin heikkoa GPS-signaalia. Ohjelmaa testattiin epäonnistumisen jälkeen ympäristössä, missä oli testaushetkellä jatkuvasti saatavilla vahva GPS-signaali, mutta jälleen ohjelman suoritus keskeytyi.

Pohdittiin, löytyisikö syy ongelmaan huonoista koordinaateista. Koordinaatit välitetään palvelimelta CloudMQTT:n kautta Android-sovellukseen string-tyyppisenä muuttujana turva-alueen ylityksen tapahtuessa. Sovelluksessa vastaanottamisen jälkeen leveys- ja pituusaste eritellään muuttujasta ja tallennetaan oikeaan muotoon (double-tyyppiseksi) ja ajetaan tarkistusfunktion läpi. Mikäli koordinaatit eivät ole määritettyjen rajojen sisällä (leveysaste > -90 ja leveysaste < 90 , pituusaste > -180 ja pituusaste < 180), droonin lennätystä ei aloiteta.

Tarkistuksen jälkeen koordinaatit lähetettiin takaisin CloudMQTT:hen, jotta palvelimen lähettämiä koordinaatteja ja droonin lennättämisessä käytettäviä koordinaatteja voitaisiin vertailla keskenään eroavaisuuksien havaitsemiseksi. Mitään eroavaisuuksia ei kuitenkaan löytynyt. Aikajanatehtävää testattiin myös kutsumalla mene määränpäähän -komentoa pelkästään korkeus asetettuna parametriksi, mutta taas ohjelman suoritus keskeytyi. Kun komento poistettiin listasta, aikaisemmin epäonnistuneet toiminnot saatiin myös suoritettua.

Ongelmaa koetettiin myös ratkaista kalibroimalla droonin IMU-järjestelmän anturit ja kompassi, mutta tälläkään ei ollut komennon suorittamiseen mitään vaikutusta. Koska ongelmaan ei saatu selvyyttä lentoon liittyvien testien avulla, ruvettiin tutkimaan Sparkin lentolokeja, jotka tallennetaan jokaisen lennätyksen yhteydessä sovellusta suorittavaan mobiililaitteeseen. Lentolokit löytyvät puhelimen sisäisestä muistista DJI-kansiosta. Lokit tallennetaan sovellukseen tekstitiedostona, jonka sisältö on salattu. Tekstitiedostot muunnettiin CSV-muotoon Airdata UAV -nimisellä sivustolla, jotta lentoon liittyviä lokeja voitaisiin tutkia Microsoft Excelillä. Myös Airdata UAV -sivustolla on mahdollista tarkastella pintapuolisesti lennätukseen liittyviä asioita, droonin anturilukemia jne. Kuva 27 havainnollistaa erästä projektin aikana tallennettua lentolokia Airdata UAV -sivustolla.

The screenshot displays the Airdata UAV web application. At the top, there is a navigation bar with links for 'UPLOAD', 'MY LOGS', 'UPGRADE', 'FORUM', and 'ABOUT'. A user account section on the right shows 'My Account' with a 'Free' status and a 'Logout' button. Below the navigation bar, there are tabs for 'FLIGHTS', 'BATTERIES', 'DRONES', 'MAINTENANCE', 'REPORTS', and 'ALERTS'. A prominent banner at the top center reads 'Hate Ads? We get it. Upgrade HD 360 members never see ads.' The main interface is divided into a sidebar on the left and a main content area. The sidebar shows an 'OVERVIEW' section with a list of flight logs, including 'Mar 9th, 2018 04:33PM' which is selected. The main content area displays the details for this flight, including a map, a list of metrics, and options to download data. The metrics are organized into categories: POWER, SENSORS, CONTROLS, and WEATHER. The map shows a flight path with a 'H' marker and a warning that there is no imagery for the location. Below the map, there are links to 'Download KML' and 'Download CSV', and an 'Add tag' field. At the bottom, there is a link to 'Add Flight Description'.

Metric / Imperial	Value
POWER	Mar 9th, 2018 04:33PM (+02:00)
SENSORS	Plane Name: Rotta
SENSORS	Flight Air Time: 00m 32s
CONTROLS	Takeoff Battery: 41%
CONTROLS	Landing Battery: 36%
WEATHER	Spark/Android DJI 4.4.1
Summary	Total Kilometrage: 4 m
Summary	Max Distance: 2 m
Summary	Max Altitude: 1.2 m
Summary	Max Speed: 0.22 m/s
Summary	Max Bat Temp: 0.00°C
Summary	Tips: 1
Summary	Warnings: 0

Kuva 27. Tarkastelun kohteena oleva lentoloki Airdata UAV -sivustolla.

Tiedoston muunnoksen jälkeen käyttäjä voi Excelin avulla perehtyä tarkemmin lennätysaikana tapahtuneisiin asioihin ja dronin eri komponenttien lukemiin. Kuva 28 havainnollistaa muunnettua tiedostoa ja osaa siitä löytyvästä datasta. Esimerkiksi noin 30 sekunnin lennätyksestä käyttäjällä on saatavilla jo useita satoja rivejä dataa tapahtumien eri hetkistä.

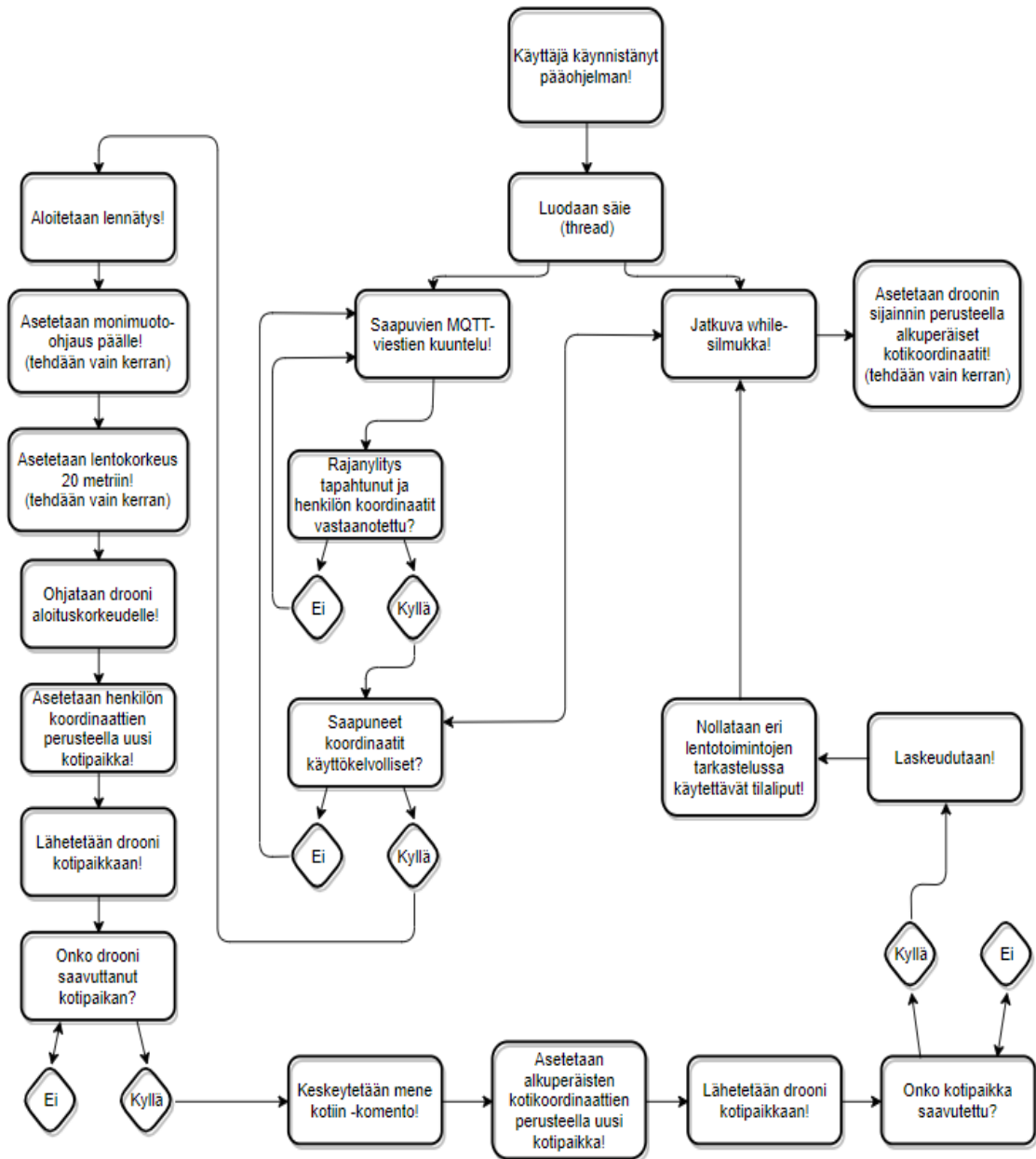
	A	B	C	D	E	F	G	H
1	latitude,longitude,altitude(feet),ascent(feet),speed(mph),distance(feet),time(
2	60.2210192980312,24.8047974420426,39.791996352084,0,0,0,40900,2018-03-09 14::							
3	60.2210192707483,24.8047975978076,39.791996352084,0,0,0,0.0299291676244,41000							
4	60.2210192379156,24.8047977251049,39.791996352084,0,0,0,0.055784778688,41100,							
5	60.2210192265328,24.8047978179425,39.791996352084,0,0,0,0.072938978712,41200,							
6	60.2210192279593,24.8047978827178,39.791996352084,0,0,0,0.08384514704,41300,2							
7	60.2210192568397,24.8047978810123,39.791996352084,0,0,0,0.080950461908,41400,							
8	60.2210192832026,24.8047978744338,39.791996352084,0,0,0,0.078537732172,41500,							
9	60.2210192839501,24.8047978579673,39.791996352084,0,0,0,0.075542325252,41600,							
10	60.2210192668742,24.804797865641,39.791996352084,0,0,0,0.07759514684,41700,20							
11	60.2210192772099,24.8047978205364,39.791996352084,0,0,0,0.0690042673,41800,20							
12	60.2210192920646,24.8047977822172,39.791996352084,0,0,0,0.061679463916,41900,							
13	60.2210192828492,24.8047977237207,39.791996352084,0,0,0,0.051340880908,42000,							
14	60.2210192594436,24.8047976886896,39.791996352084,0,0,0,0.0468585973,42100,20							
15	60.2210192649771,24.8047975897587,39.791996352084,0,0,0,0.0293581374224,42200							
16	60.2210192763615,24.8047975003839,39.791996352084,0,0,0,0.0132010830996,42300							
17	60.2210192930531,24.8047973821187,39.791996352084,0,0,0,0.0110093179376,42400							
18	60.2210192991744,24.8047972918815,39.791996352084,0,0,0,0.027213091422,42500,							

Kuva 28. Osa CSV-muotoon muunnetun lentolokin tekstitiedostosta Microsoft Excelissä.

Mene määränpähän -komennon toimimattomuus sovelluksessa ei selvinnyt edes lentolokia tutkimalla. Edellytykset lennätykselle olivat olemassa (vahva GPS-signaali jne.), mutta komentoa ei saatu suoritettua. Viimeisenä ratkaisuna ongelmaa lähdettiin tiedustelemaan DJI:n kehitystuesta (sähköpostikeskustelu käytiin 12.3.–14.3.2018). Tuesta selvisi, että komento ei ole käytettävissä DJI Sparkilla, koska drooni ei tue reittipiste-tehtäviä (tämä oli entuudestaan tiedossa), mikä tarkoittaa, että mene määränpähän -komennon käyttämistä ei siis myöskään tueta, sillä se koostuu peruserrokseltaan reittipiste-tehtävästä (tämä asia ei ollut tiedossa). Kehitystuki myös pahoitteli tämän tiedon puuttumista dokumentaatiosta. Ongelmaan tiedusteltiin lisäksi vaihtoehtoisia ratkaisua mutta vastausta ei koskaan saatu.

Vaihtoehtoinen toteutustapa

Koska alkuperäinen toteutustapa mene määränpähän -komennon hyödyntämisestä osana aikajanatehtävää kariutui kehitystuesta saatuun palautteeseen, otettiin vaihtoehtoinen tapa käyttöön. Vaihtoehtoiseen ratkaisuun kuului mene kotiin -komennon käyttäminen ja uuden kotipaikan määrittäminen droonille sen lähtöpaikan ja henkilön koordinaattien perusteella. Kuva 29 havainnollistaa Android-sovellukseen luotua ohjelman viimeisintä versiota.



Kuva 29. Lennättämiseen liittyvien toimintojen kulku Android-sovelluksessa.

Kuvassa 29 tapahtuu tiivistettynä seuraavat asiat: Android-sovellus odottaa pääohjelman käynnistämisen jälkeen palvelimelta saapuvia viestejä, jotka välitetään Cloud-MQTT:n kautta. Kun palvelin on lähettänyt turvarajan ylityksestä ilmoituksen ja sen jälkeen henkilön koordinaatit, vastaanotetut leveys- ja pituusasteet ajetaan tarkistusfunktion läpi (samaa funktiota käytetään kotikoordinaattien tarkistukseen). Jos koordinaatit ovat aikaisemmin mainittujen rajojen sisällä, käynnistetään lennättämiseen liittyvien komentojen suoritus. Komennot suoritetaan kuvan mukaisessa järjestyksessä ja jokaisen

toiminnon suoritusta valvotaan vastaavilla tilalipuilla. Seuraavaa komentoa ei aloiteta, ennen kuin ilmoitus edellisen suorituksesta on vastaanotettu.

Lennättämiseen liittyvien toimintojen seurantaan käytetään tilalippujen lisäksi vielä lentosäädintila-luokan tilatarkistuksia apuna. Luokan avulla prototyypissä tarkkaillaan mene kotiin -komennon suoritusta nimenomaan tilanteessa, jossa droni on lentämässä henkilön koordinaatteihin. Mene kotiin -komento sisältää dronin laskeutumisen maahan, ja tämä toiminto halutaan estää, kun henkilön luokse on saavuttu. Lentosäädintilan tilatarkkailun ilmoitettua, että laskeutuminen on aloitettu, komennon suoritus keskeytetään ja dronille asetetaan alkuperäisten kotikoordinaattien mukaan uusi kotipaikka. Määränpäähän laskeutumisen jälkeen ohjelma jää odottamaan uutta rajanylitystä.

Vaihtoehtoisen toteutustavan aikana esiintyneet ongelmat ja niiden ratkaisut

Vaihtoehtoisen tavan kehitysvaiheessa ilmeni useita eri ongelmia. Ensimmäisessä versiossa lennättämiseen liittyviä toimintoja saatiin testattua vaihtelevalla menestyksellä. Ongelmana oli, että ohjelma "jäätty" hetkellisesti komentojen suorituksen aikana, ja jääntymisen seurauksena osa toiminnoista joskus suoritettiin ja joskus ei. Ongelman aiheuttajaksi selvisi pääohjelman funktioiden kutsuminen MQTT:n takaisinkutsu-metodin sisällä kohdassa, jossa saapuneita viestejä tarkistetaan, mikä puolestaan aiheutti ohjelman lukkiutumisen. Lukkiutuminen saatiin estettyä siirtämällä funktioiden kutsuminen pääohjelmaan.

Seuraavassa versiossa MQTT:n takaisinkutsu, kotikoordinaattien ja henkilön koordinaattien tarkistus sekä lennätyksen komentoja suorittavan funktion kutsuminen (sama, jota kutsuttiin aikaisemmin MQTT:n takaisinkutsun sisällä) siirrettiin omaan säikeeseensä pääohjelmassa rakenteen selkeyttämiseksi. Tehtyjen muutosten seurauksena droni nousi aloituskorkeuteen, 1,2 metriin, mutta ei saanut asetettua henkilön koordinaatteja uudeksi kotipaikaksi, mistä seurasi myös virheilmoitus. Syyn epäiltiin jälleen olevan koordinaateissa, mutta tarkistuksen jälkeen ongelman todettiin sijaitsevan jossain muualla. Hämmennystä aiheutti entisestään se, että ensimmäisen testiversioiden aikana lennätykseen liittyviä komentoja saatiin suoritettua vaihtelevalla menestyksellä. Syyksi havaittiin komentoja suorittavan funktion kutsuminen säikeestä tavalla, jossa kaikkia toimintoja kutsutaan kertaalleen tarkistamatta niiden suorituksen etenemistä ja suoritusjärjestystä mitenkään.

Myös ensimmäisessä versiossa komentoja kutsuttiin tällä tavalla. Selvisi, että ohjelman suorituksessa esiintyy ristiriitoja: esimerkiksi drooni haluaa lentää mene kotiin -komennolla henkilön luokse, ennen kuin aseta kotikoordinaatit -komentoa on ehditty suorittaa loppuun, mikä taas aiheuttaa virheilmoituksen mene kotiin -komennossa. Ensimmäisessä versioissa lennätykseen liittyviä toimintoja saatiin testattua täysin sattumalta ohjelman suorituksessa ilmenneiden viiveiden takia. Ongelmaa yritettiin ratkaista lisäämällä while-silmukka komentojen suorituksen ympärille ja silmukasta poistuttaisiin vasta, kun viimeinen toiminto suoritettu. Tämän seurauksena aiheutui koko sovelluksen kaatuminen silmukan aiheuttaman viivästymisen takia pääohjelmassa. Syy sovelluksen kaatumiseen on silmukan suorittaminen käyttöliittymä-säikeessä, jolloin käyttöliittymä ei pysty tekemään muuta kuin käymään silmukkaa läpi. Silmukka on asetettava erilliseen säikeeseen, jotta käyttöliittymä-säie voi olla vapaasti käytössä. (23.)

Viimeisimmässä versioissa lennätykseen liittyvät komennot siirrettiin aikaisemmin luotuun säikeeseen jatkuvan while-silmukan sisälle. Komennolle lisättiin myös tilaliput toimintojen etenemisen rajoittamiseksi niin, että seuraavaa ei suoriteta, ennen kuin ilmoitus edellisen suorituksesta on vastaanotettu. Tällä tavalla kaikki komennot saatiin suoritettua, mutta droonia ei saatu silti lentämään henkilön luokse. Ongelman syyksi selvisi, että ilmoitus suorituksesta tulee silloin, kun vuorossa oleva komento aloitetaan, eikä silloin, kun se lopetetaan. Tästä aiheutui, että ennen henkilön luokse lentämistä drooni ehti jo laskeutua takaisin alkuperäiseen kotipaikkaansa ohjelman suorituksen mukaisesti.

Ohjelman suoritusta varten lisättiin viive, jonka avulla drooni saatiin vihdoinkin lentämään kohti henkilöä, mutta myös tästä toteutustavasta seurasi ongelmia. Tällä tavalla drooni ennättää lentämään henkilön luokse viiveeseen määritetyn ajan verran, ennen kuin se kääntyy takaisin kohti alkuperäistä kotipaikkaansa. Tämä toteutustapa ei ollut vartenotettava, koska henkilön sijainti vaihtelee jokaisessa lennätyksessä. Jos aika määritetään liian pitkäksi, drooni ehtii laskeutua maahan henkilön luona mene kotiin -komennon mukaisesti, ja kyseistä tilannetta ei haluta prototyypissä tapahtuvan. Tarkoituksena on, että drooni käy määränpäässä ja palaa takaisin. Tilanteen välttämiseksi käyttöön otettiin myös lentosäädintila-luokan tilatarkistustoiminnot.

Tilatarkistustoiminnon ja komentojen tilalippujen avulla droonin etenemistä pystyttiin seuraamaan halutulla tavalla, joka menee seuraavasti: alkuasetusten jälkeen (monimuoto-ohjaus, lentokorkeus) drooni nostetaan aloituskorkeudelle, asetetaan määrän-

pään koordinaatit ja lähetetään alus henkilön luokse. Määränpäähän saavuttua tilatar-
kistustoiminnon avulla seurataan hetkeä, jolloin drooni lähtee laskeutumaan. Laskeutu-
misen alettua mene kotiin -komennon suoritus keskeytetään ja asetetaan alkuperäinen
lähtöpaikka uudeksi kotipaikaksi. Kun määränpää on saavutettu, laskeudutaan maahan.
Lennätyksen aikainen lentokorkeus vaihtelee nykyisellä toteutustavalla mene kotiin -ko-
mennon mukaisesti seuraavasti: Mikäli henkilön ja dronin välinen etäisyys on suurempi
kuin 20 metriä, lähestytään henkilöä 20 metrin korkeudella. Jos etäisyys on alle 20 metriä
mutta yli 3 metriä, lähestytään henkilöä 2,5 metrin korkeudella. Jos etäisyys on alle 3
metriä, drooni laskeutuu suoraan alas. Tämä toimintamalli on siis prototyypin viimeisin
ja voimassa oleva versio.

Rajoituksia nykyisessä toteutustavassa

Prototyypissä etäisyys dronin lennättämisen suhteen on rajoittunut 30 metriin käytettä-
vän mene kotiin -toiminnon takia. Mikäli henkilö on tätä matkaa etäämmällä, uusia koor-
dinaatteja ei saada tallennettua laitteelle, ja tästä seuraa myös virheilmoitus. Jos dronin
lentoetäisyyttä haluttaisiin kasvattaa, yhtenä ratkaisuna voisi olla esimerkiksi seuraavan-
lainen toteutustapa: Dronin ja henkilön välinen etäisyys laskettaisiin molempien koor-
dinaattien perusteella. Mikäli koordinaattien välinen etäisyys olisi suurempi kuin 30 metriä,
laitteen ja henkilön väliin muodostuvalta kuvitteelliselta janalta voitaisiin tallentaa sopivin
välimatkoin uusi kotipaikka, johon drooni lähetetään.

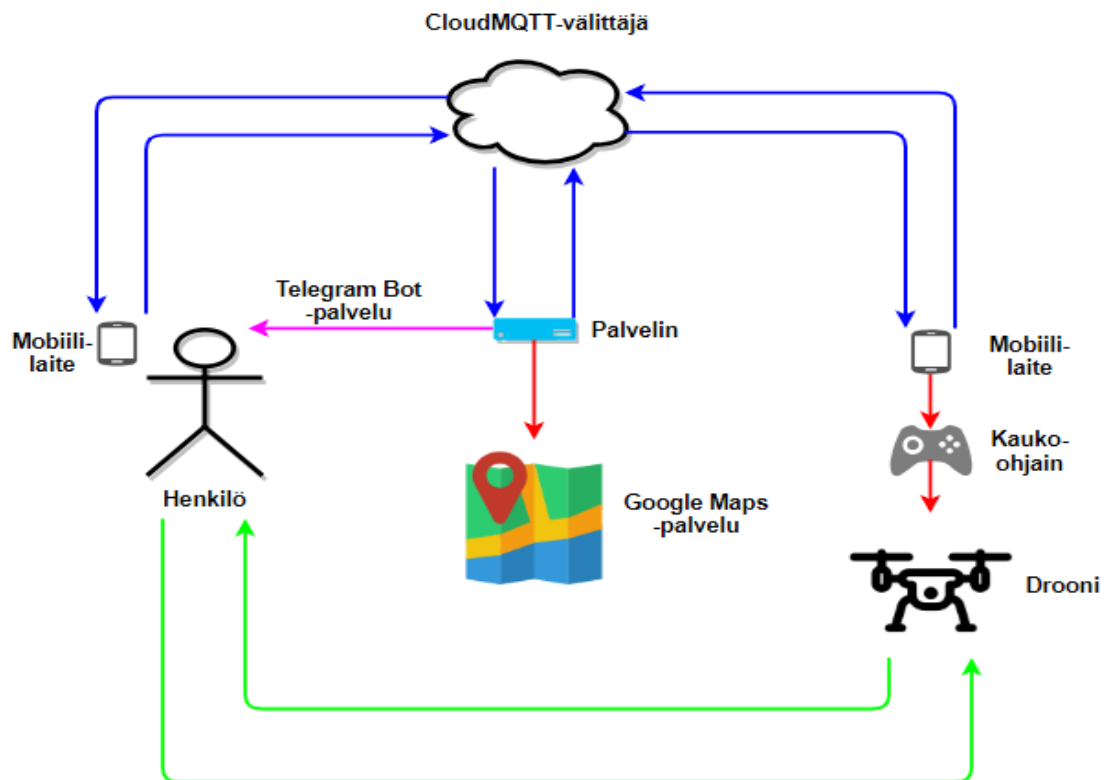
Tätä tapaa jatkettaisiin siihen asti, kunnes etäisyys näiden kahden välillä olisi sallittujen
rajojen sisällä, jolloin henkilön koordinaatit voitaisiin tallentaa kotipaikaksi. Määränpää-
hän saavuttua drooni lähetettäisiin takaisin alkuperäiseen lähtöpaikkaansa. Kotipaikan
30 metrin rajoitus koskee lennättämisen aikana mobiiliohjelmistokehityspaketin kautta
päivitetyjä kotipaikkoja. Droni voi palata alkuperäiseen kotipaikkaansa automaattisesti.
(12.) Rajallisen, käytettävissä olevan ajan takia tätä toteutustapaa ei kuitenkaan kehi-
tetty. Nykyisellä tavalla automatisoituja toimintoja pystytään havainnollistamaan ilman,
että käyttäjän tarvitsee olla itse sovelluksen kanssa tekemisissä.

3.7 Lopputulos

Projektin viimeistely versio kokonaisuudessaan sisältää siis seuraavat asiat: henkilön ja
henkilön mukana olevan mobiililaitteen, jonka avulla sijaintitietoja kerätään. Henkilö voi

myös määrittää mobiililaitteen sijainnin perusteella keskipisteen ja säteen (säteen pituus määritetään metreinä), jotka yhdessä muodostavat tarkkailtavan turva-alueen. Kerätty data välitetään palvelimelle. Datan välittäjänä toimii CloudMQTT-pilvipalvelu, jonka kautta kerätty tieto toimitetaan palvelimelle ja droonia ohjaavalle mobiililaitteelle. Palvelimena projektissa toimii Raspberry Pi 3 B -mikrokontrolleri, joka kuuntelee, vastaanottaa ja lähettää dataa tarpeen mukaan. Palvelimelle on lisäksi luotu sivusto, missä kerätty data näytetään kartalla käyttämällä Google Maps -palvelua. Sivustolla näytetään henkilön määrittelemä turva-alue, vastaanotetut sijaintitiedot ja droonin sijaintitiedot. GPS toimii henkilön ja droonin sijaintitietojen lähteenä. Palvelimella suoritetaan myös Telegram Bot -palvelua, joka ilmoittaa viestien välityksellä turva-alueen ylityksistä ja rajojen sisälle palaamisista tarkkailun kohteena olevalle henkilölle.

Henkilön ylittyä turva-alueen palvelin lähettää tapahtumasta tiedon droonia ohjaavalle mobiililaitteelle, joka aktivoi lennättämiseen liittyvät toiminnot. Lennättämisen aikana droonin sijaintitiedot välitetään palvelimelle. Kun henkilön luona on käyty ja palattu takaisin lähtöpisteeseen, jäädytään odottamaan uutta rajanylitystä. Kuva 30 havainnollistaa kehitettyä kokonaisuutta.



Kuva 30. Prototyypin kokonaisuus.

4 Huomioitavaa jatkokehitystä ajatellen

Mikäli turva-alueen toteutusta lähdetäisiin kehittämään esimerkiksi tuotantovaiheeseen asti, on syytä tarkastella jatkokehitykseen mahdollisesti vaikuttavia asioita. Prototyypissä käytettävä droni on pieni ja kevytrakenteinen (massa on 300 g), mikä tekee siitä erittäin vaikutusalttiin esimerkiksi tuulenpuuskille (testien aikana tämä havaittiin useaan kertaan). Dronin lentoaika on laitteistodokumentaation mukaan 16 minuuttia, joka saattaisi olla riittävä, mikäli käyttäjä määrittäisi turva-alueen suhteellisen lähelle. Tilanteessa, jossa turva-alueen etäisyys olisi esimerkiksi joitakin kilometrejä, ei prototyypissä käytettyä laitetta voisi hyödyntää. Wi-Fi-verkon signaalin kantavuus kauko-ohjaimen ja Sparkin välillä on Euroopassa 500 metriä (Yhdysvalloissa kantavuus on 2 kilometriä) eli tätä pidemmälle ei signaalin puitteissa pysty tai kannata lentää, jotta yhteys laitteiden välillä ei katoa.

Lennätyksessä on myös huomioitava paluumatkaan kuluva aika, joka riippuu dronin ja lähtöpisteen välisestä etäisyydestä ja lennätysvetkellä vallitsevasta säästä. Tästä seuraa, että droni ei välttämättä kykene lentämään henkilön luokse, koska akun varaus on laskenut liian alhaiseksi ja on palattava takaisin lähtöpisteeseen. Riittävä aika kotiinpaluuta varten pystytään varmistamaan akun varausta tarkkailevalla, DJI:n ohjelmoitavalla toiminnolla (Low Battery RTH). Käyttäjä voi määrittellä akun varaukseen liittyvän kynnyksen (tyypillisesti 25–50 %), jonka alittuessa droni voidaan laittaa palaamaan takaisin lähtöpisteeseensä. Vastaavanlainen turvamekanismi (Failsafe RTH) on myös käytettävissä tilanteessa, jossa yhteys kauko-ohjaimen ja dronin välillä katoaa yli kolmeksi sekunniksi, jolloin droni palaa takaisin kotipisteeseensä. Turvamekanismin käyttö edellyttää onnistunutta kotipaikan tallennusta ja kompassin normaalia toimintaa. (12.)

Projektissa käytetyllä Spark-droonilla ei ole käytettävissä reittipiste-, kuuma piste- tai seuraa minua -tehtäväoperaattoreita (Follow Me), joiden avulla henkilön paikantaminen onnistuisi huomattavasti helpommin kuin keinoilla, joita prototyypin toteutuksessa käytettiin (operaattoreilla dronin voi suoraan lähettää haluttuihin koordinaatteihin tai asettaa sen seuraamaan haluttuja koordinaatteja jatkuvasti). Jatkokehityksen kannalta suositeltavampaa olisikin käyttää rakenteeltaan raskaampaa droonia, joka ei ole yhtä altis tuulenpuuskille ja jossa on mahdollista käyttää edellä mainittuja tehtäväoperaattoreita ja lentoaika olisi suurempi (esimerkiksi kuluttajakäytössä olevat Phantom-sarjan tuotteet ja Mavic Pro).

Huomioitavana ovat myös muut ilmastosta aiheutuvat tekijät kuten Suomessa talviaikana kylmyys. Valtaosa DJI:n tuotteiden käyttölämpötiloista toimii välillä 0–40 °C, kuten esimerkiksi kuluttajakäytössä olevat tuotteet mukaan luettuna projektissa käytettävä Spark. Poikkeuksina DJI:lta löytyy ammattikäyttöön tarkoitettu Inspire 2 (-20–40 °C) ja yrityskäyttöön tarkoitetut Matrice 200 series (-20–40 °C) ja Matrice 600 Pro (-10–40 °C). Haittapuolena on kuluttajakäytössä oleviin laitteisiin verrattuna huomattavasti korkeampi hinta. Esimerkiksi projektissa käytettävän Sparkin saa DJI:lta tilattua hintaan 499 € (ilman kauko-ohjainta), kun taas yrityskäyttöön tarkoitettu Matrice 600 Pro -droni maksaa 5 699 €.

Vaihtoehtoisena ratkaisuna voisi olla esimerkiksi varustaa kuluttajakäytössä oleva droni lämpöhuovalla tai avaruuspeitteellä (materiaali, joka on alun perin kehitetty NASA:n Apollo-ohjelmalle (24)) lämmön eristämiseksi. Ratkaisu saattaisi parantaa mahdollisuuksia lennättää laitetta kylmemmällä säällä, ja se auttaisi säästämään tuotteen hankintakustannuksissa. Lisäksi dronin lähtöalustaan liittyvät asiat täytyisi suunnitella tarkasti. Lähtöalustan tulisi olla sisällä lämmitetyssä tilassa (varsinkin talvella), ja siellä tulisi olla tarvittava virtalähde latausalustaa varten, jotta droonia saadaan ladattua, kun saavutaan takaisin ”kotipaikkaan”.

Jatkuva virtalähde tarvittaisiin myös kauko-ohjaimelle ja mobiililaitteelle. Lähtöpaikan rakennuksen seinämien tulisi olla materiaalia, joka aiheuttaa mahdollisimman vähän häiriöitä Wi-Fi-verkkoon (puulla, lasilla ja synteettisellä materiaalilla on pieni häiriön mahdollisuus, laastilla ja sementillä suuri ja metallilla erittäin suuri häiriön mahdollisuus (25)). Lähtöalusta itsessään tulisi olla sijoitettuna mahdollisimman katveettomalle alueelle parhaimman signaalin mahdollistamiseksi laitteiden välillä.

Kaiken edellä mainitun lisäksi on vielä huomioitava lennättämiseen liittyvät määräykset Suomessa ja kolmen vuoden siirtymäajalla voimaan astuvat EU:n määräykset. Tuotannossa olevan dronin määräyksinä toimisivat tästä raportista poiketen kauko-ohjatun ilma-aluksen lennätyksessä sovellettavat ilmailumääräykset, koska droni suorittaisi lentotyötä. Prototyypin jatkokehitystä varten soitettiin Trafille (puhelinkeskustelu käytiin 17.4.2018). Trafilta tiedusteltiin, olisiko automatisoidun dronin käyttäminen mahdollista ilman suorassa näköyhteydessä olevaa tarkkailijaa tavalla, jossa kamerasta välitetään reaaliaikaista kuvaa järjestelmään, josta käyttäjä seuraa lennon etenemistä. Trafilta saadun vastauksen mukaan tämä on mahdollista mutta edellytyksenä on, että käyttäjä lähettää aiheesta anomuksen Trafille, joka puolestaan tekee ilmatilan varauksen alueelle,

jossa droonia on tarkoitus lennättää. Anomus on myös uusittava Trafín kanssa määräajoin. Lennätystä varten järjestelmässä on lisäksi oltava hätämekanismi, jonka avulla alus saadaan laskeutumaan tai palaamaan takaisin lähtöpaikkaan. Jatkokehityksen kannalta voidaan todeta, että droonien toimintojen ja määräysten puitteissa prototyyppiä on siis mahdollista kehittää haluttaessa pidemmälle.

5 Yhteenveto

Insinööriyössä kehitettiin prototyyppi järjestelmästä, jonka avulla maastossa liikkuvan henkilön kunto voidaan käydä tarkistamassa käyttäjän määritettävissä olevan turva-alueen ylityksen tapahduttua. Android-käyttöjärjestelmällä varustetulle mobiililaitteelle luotiin ohjelma, joka kuuntelee ja vastaanottaa palvelimelta saapuvia, turva-alueen ylitykseen liittyviä viestejä ja hallinnoi droonin lennättämisessä käytettäviä toimintoja. Kunnan tarkastamista varten käytettiin kameralla varustettua DJI Spark -droonia, joka välittää lennätysten aikana reaaliaikaista kuvaa mobiililaitteelle. Käyttäjän (esimerkiksi hoitohenkilökunnan) päivittäisten toimintojen helpottamiseksi lennättämiseen liittyvät komennot automatisoitiin. Ylityksen tapahduttua henkilön koordinaatit tallennetaan, ja mikäli vastaanotetut leveys- ja pituusasteet ovat käyttökelpoiset, aktivoidaan drooni. Aktivoinnin tapahduttua alus lennätetään henkilön luokse, minkä jälkeen palataan takaisin lähtöpisteeseen ja jäädään odottamaan uutta rajanylitystä. Lennätysten aikana droonin sijaintitiedot välitetään palvelimelle.

Työn toinen kehittäjä loi henkilön paikantamisessa käytettävän järjestelmän ja palvelimen, joka näyttää eri osapuolten koordinaatit ja asetetun turva-alueen Google Maps -palvelua hyödyntävällä sivustolla. Palvelin välittää lisäksi rajan ylitykseen liittyviä viestejä henkilölle käyttämällä Telegram Bot -palvelua. Koordinaattien ja turva-alueeseen liittyvien tietojen välityksessä käytettiin CloudMQTT-palvelua.

Prototyypin toteutuksen aikana esiintyi useita eri ongelmia, jotka saatiin myös vaihtoehtoisilla toteutustavoilla ratkaistua. Kehityksessä pääongelmat liittyivät droonin lennättämisessä käytettäviin komentoihin ja niiden suoritustapaan Android-sovelluksessa. Reaaliaikaisen kuvan striimaamista toiselle päätelaitteelle ei ajan puitteissa ehditty kehittää, vaikka mobiilidatan ja Wi-Fi-verkon yhtäaikaiselle käytölle löydettiin vaihtoehtoinen

ratkaisu Speedify-sovelluksesta. Lopputuloksena kuitenkin saatiin kehitettyä lentoetäisyyksiltään rajoitettu ja automatisoitu järjestelmä, jonka avulla turva-alueprototyyppiä on mahdollista havainnollistaa.

Lähteet

- 1 History of Drones. Verkkoaineisto. Dronepedia. <<https://dronepedia.xyz/the-history-of-drones/>>. Luettu 22.10.2017.
- 2 Horror bomb attacks in Venice. 2016. Verkkoaineisto. AlterVista. <<http://veniceok.altervista.org/horror-bomb-attacks-in-venice/>>. 27.2.2016. Luettu 21.10.2017.
- 3 Hewitt-Sperry Automatic Airplane. 2017. Verkkoaineisto. Wikipedia. <https://en.wikipedia.org/wiki/Hewitt-Sperry_Automatic_Airplane>. Päivitetty 8.4.2017. Luettu 21.10.2017.
- 4 Long-Range Gun with Engine (RAE Larynx) (UK, 1925–27). Verkkoaineisto. UAV universe. <<https://sites.google.com/site/uavuni/1910-s>>. Luettu 22.10.2017.
- 5 History of quadcopters and other multirotors. Verkkoaineisto. Krossblade Aerospace Systems LLC. <<http://www.krossblade.com/history-of-quadcopters-and-multirotors/>>. Luettu 22.10.2017.
- 6 BCW. 2017. Bolt Drones: The Future of First Person View Flying. Verkkoaineisto. Best Crowdfunding Websites. <<http://best-crowdfunding-websites.com/bolt-drones-the-future-of-first-person-view-flying/>>. 15.5.2017. Luettu 17.4.2018.
- 7 Spark. Verkkoaineisto. DJI Developer, Products. <<https://developer.dji.com/products/>>. Luettu 13.4.2018.
- 8 AirLink. 2017. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <<https://developer.dji.com/mobile-sdk/documentation/introduction/component-guide-airlink.html>>. 27.6.2017. Luettu 26.1.2018.
- 9 HD Wi-Fi Video transmission. Verkkoaineisto. DJI Store, Discounts. <<https://store.dji.com/product/spark?site=developer&vid=24661>>. Luettu 13.4.2018.
- 10 Remote Controller. 2017. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <<https://developer.dji.com/mobile-sdk/documentation/introduction/component-guide-remotecontroller.html>>. 27.6.2017. Luettu 26.1.2018.
- 11 Mobile SDK Introduction. 2017. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <https://developer.dji.com/mobile-sdk/documentation/introduction/mobile_sdk_introduction.html>. 27.6.2017. Luettu 29.1.2018.
- 12 Flight Controller. 2017. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <<https://developer.dji.com/mobile-sdk/documentation/introduction/component-guide-flightController.html>>. 27.6.2017. Luettu 26.1.2018.

- 13 Henttu, Pekka & Metsävainio, Päivi. 2016. Kauko-ohjatun ilma-aluksen ja lennokin lennättäminen. Verkkoaineisto. Trafi Liikenteen turvallisuusvirasto. <https://www.trafi.fi/file-bank/a/1482415412/c34a1bef37860a2559d61acf4fdebb3a/23514-OPS_M1-32_VALMIS_maarays_RPAS_fi.pdf>. 23.12.2016. Luettu 28.1.2018.
- 14 Onko omistamani laite kauko-ohjattu ilma-alus vai lennokka? Verkkoaineisto. Trafi Liikenteen turvallisuusvirasto, Tietopalvelut, Usein kysyttyä. <https://www.trafi.fi/tietopalvelut/usein_kysyttya/ilmailu_-_miehittamattomat_ilma-alukset_ja_lennokit>. Luettu 28.1.2018.
- 15 Reinikka, Juha. 2017. EASA Määräysluonnos ja omatekoiset lennokit. Verkkoaineisto. Lentsu Lennokiturvallisuus. <<http://www.lentsu.fi/easa-maaraysluonnos-ja-omatekoiset-lennokit/>>. 6.5.2017. Luettu 29.1.2018.
- 16 Integrate SDK into Application. 2017. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <<https://developer.dji.com/mobile-sdk/documentation/application-development-workflow/workflow-integrate.html>>. 29.12.2017. Luettu 30.1.2018.
- 17 Creating a Camera Application. 2018. Verkkoaineisto. DJI Developer, Mobile SDK, Documentation. <<https://developer.dji.com/mobile-sdk/documentation/android-tutorials/index.html>>. 2.1.2018. Luettu 3.2.2018.
- 18 DJI Diana. 2018. DJI GO 4 live streaming. Verkkoaineisto. DJI Forum, Spark. <<https://forum.dji.com/thread-128430-1-1.html>>. 9.1.2018. Luettu 9.2.2018.
- 19 Syahidillah, Wildan Maulana. 2017. MQTT Android Client Tutorial. Verkkoaineisto. Wildan's tech blog. <<https://wildanmsyah.wordpress.com/2017/05/11/mqtt-android-client-tutorial/>>. 11.5.2017. Luettu 13.2.2018.
- 20 What is Channel Bonding? Verkkoaineisto. Speedify. <<http://speedify.com/features/channel-bonding/>>. Luettu 10.2.2018.
- 21 Two connections are better than one. Verkkoaineisto. Speedify. <<http://speedify.com/>>. Luettu 13.4.2018.
- 22 class MissionControl. Verkkoaineisto. DJI Developer, Mobile SDK, API Reference. <https://developer.dji.com/api-reference/android-api/Components/Missions/DJIMissionControl.html?search=starttimeline&i=0&#djmmissioncontrol_starttimeline_inline>. Luettu 26.1.2018.

- 23 Daghestani, Zaid. 2011. Implementing a while loop in android. Verkkoaineisto. Stack Overflow, Questions. <<https://stackoverflow.com/questions/7478941/implementing-a-while-loop-in-android>>. 20.9.2011. Luettu 1.3.2018.
- 24 Avaruuspeite. Verkkoaineisto. Varmuusvarasto.fi. <<http://www.varmuusvarasto.fi/avaruuspeite.html>>. Luettu 13.4.2018.
- 25 Wi-Fi- ja Bluetooth-yhteyksien mahdolliset häiriöiden aiheuttajat. 2016. Verkkoaineisto. Apple, Tuki. < <https://support.apple.com/fi-fi/HT201542>>. Luettu 13.4.2018.