

LATVIJAS UNIVERSITĀTES  
DATORIKAS FAKULTĀTE

**SABIEDRĪBAS VIRZĪTA MOBILĀ LIETOTNE  
COVID-19 INFEKCIJAS RISKĀ MAZINĀŠANAI**

BAKALĀURA DARBS

Autors: Edgars Baumanis

Studenta apl. nr. - eb17079

Vadītājs: Dr. dat.zin. Leo Seļāvo

**RĪGA 2021**

# Anotācija

Bakalaura darba autors Edgars Baumanis izvēlējies tēmu “Sabiedrības virzīta mobilā lietotne Covid-19 infekcijas riska mazināšanai”, atsaucoties uz vīrusa Covid-19 izplatību Latvijā un pasulē un projekta “Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project)”[4] monitorēšanas risinājuma nepieciešamību. Monitorēšana iekļauj telpas cilvēku skaita ievākšanu, dalībnieku apziņošanu un telpas datu nosūtīšanu uz kopēju datubāzi.

Pētījuma realizēšana ir aktuāla darba rakstīšanas laikā, jo Covid-19 turpina ierobežot visu pasauli. Bakalaura darba mērķis ir izpētīt esošas sabiedrības vadītas lietotnes, dažādus cilvēku monitorēšanas risinājumus, kā arī izplānot un realizēt risinājumu, kas atbilst projekta vajadzībām.

**Atslēgvārdi:** Firebase, Android, iBeacons, monitorēšana, BLE.

# Abstract

## COMMUNITY DRIVEN MOBILE APPLICATION TO REDUCE THE RISK OF COVID-19 INFECTION

The author of the bachelor's thesis Edgars Baumanis has chosen the topic "Community driven mobile application to reduce the risk of Covid-19 infection" referring to the spread of Covid-19 virus in Latvia and the projects "Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project)"[4]. Monitoring includes gathering the number of people in the room, notifying the participants and sending the room data to a common database.

The implementation of the research is relevant at the time of writing, as Covid-19 continues to limit the whole world. The aim of the bachelor's thesis is to study existing community driven applications, various human monitoring solutions, as well as to plan and implement a solution that meets the needs of the project.

**Keywords:** Firebase, Android, iBeacons, monitoring, BLE.

# Saturs

<b>Anotācija</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Saturs</b>	<b>4</b>
<b>Apzīmējumu saraksts</b>	<b>6</b>
<b>Ievads</b>	<b>7</b>
<b>1. Projekts un ieguldījums</b>	<b>8</b>
1.1. Problēmas apraksts	8
<b>2. Esošu risinājumu izpēte</b>	<b>9</b>
2.1. Citas sabiedrības virzītas sistēmas	9
2.1.1. Waze	9
2.1.2. TikTok	10
2.2. Cilvēku monitorēšana telpā	11
2.2.1. Digitālo attēlu analizēšana	11
2.2.3. Bluetooth BLE bākas	13
2.2.3.1 iBeacons	14
2.2.3.2 Eddystone	15
2.2.4. CO2 mērījumu veikšana	18
<b>3. Izvēlētās tehnoloģijas</b>	<b>20</b>
3.1. Mobilais telefons	20
3.2. Firebase	21
3.2.1. Firebase autentifikācija	21
3.2.2. Firebase tiešsaistes datubāze	21
3.2.3. Firebase mākoņfunkcijas	22
3.3. iBeacons protokola BLE bāka	22
<b>4. Realizētais risinājums</b>	<b>23</b>
4.1. Tehniskie risinājumi	23
4.1.1. Firebase (Back-end)	23
4.1.1.1. Firebase autentifikācija	23
4.1.1.2. Firebase tiešsaistes datubāze	24
4.1.1.3. Firebase mākoņfunkcijas	25
4.1.2. Android mobila lietotne (Front-end)	28
4.1.2.1. Ekrānskati un dizains	28
4.1.2.2. Strādāšanas principi	31
4.1.3. Bluetooth BLE bākas (Sensori)	31

<b>5. Rezultāti</b>	<b>34</b>
5.1. Testēšana	34
<b>Secinājumi</b>	<b>36</b>
<b>Izmantotā literatūra</b>	<b>37</b>

# Apzīmējumu saraksts

Github - Repozitoriju glabātuve.

ID - Identifikators.

API - Funkciju un procedūru kopums, kas ļauj izveidot lietojumprogrammas, kas piekļūst operētājsistēmas, lietojumprogrammas vai cita pakalpojuma funkcijām vai datiem.

MQTT - Ziņojumapmaiņas protokols, kas pielieto abonētājus un publicētājus.

UUID - Universāli unikāls identifikators.

BLE - *Bluetooth* zemas jaudas

Tx power - pāraides jauda

SDK - Programmatūras izstrādes rīks

HTTPS - *Hypertext Transfer Protocol Secure* jeb komunikācijas protokols drošai saziņai datortīklā.

GPS - Globālā Pozicionēšanas Sistēma

ML - Mākslīgais intelekts

JSON - JavaScript Objektu Notācija

OS - Operētājsistēma

# Ievads

Covid-19 pandēmija ir skārusi visu pasauli un turpina ietekmēt katra cilvēka ikdienu. Lai mazinātu Covid-19 izplatību Valsts Pētījumu Programma IT programmā realizē “Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project)”[4]. Projektā nepieciešams izveidot risinājumu cilvēku monitorēšanai. Risinājumam jāievēro dalībnieku privātums, taču jānodrošina, ka dalībnieku privātums tiek saglabāts.

Darba mērķis ir rast risinājumu projekta monitorēšanas vajadzībām. Tādēļ tiek nodefinēti šādi apakšmērķi:

- Izpētīt populāras sabiedrības virzītas lietotnes, lai smeltos iedvesmu un mācītos no jau strādājoša produkta;
- Izpētīt dažādus veidus kā veikt cilvēku uzskaiti telpā;
- Realizēt un dokumentēt izvēlēto risinājumu;
- Veikt realizētā risinājuma testēšanu.

Lai apziņotu cilvēkus par apkārtējās vides kvalitāti tiek izmantoti vides sensori, taču šī darba ietvaros tie netiek apskatīti un to dati tiek iegūti no ārējas datubāzes, pielietojot Jura Veldres izstrādātu, API.

Bakalaura darba ietvaros tiek sniegts ieskats sistēmas izstrādē, kuras izveidē izmantotie risinājumi ir paredzēti lai lietotājam sniegtu maksimālu privātumu, pēc iespējas ērtāku lietotājpieredzi un augstas kvalitātes risinājumu.

# 1. Projekts un ieguldījums

Sakarā ar Covid-19 izplatību Latvijā un pasaulē atbildīgās valstu iestādes ir uzņēmušās sabiedrībai ieviest vadlīnijas, kuras nepieciešams ievērot, lai mazinātu vīrusa Covid-19 vīrusa izplatību. Dažas no šīm vadlīnijām ir masku nēsāšana, intensīvāka roku mazgāšana un divu metru ievērošana. Sekojot divu metru ievērošanas noteikumam, iestādēs tika pieņemts cilvēku uzturēšanās limits telpā. Limits atkarīgs no telpas iespējamās staigājamās teritorijas, taču parasti limits tiek rēķināts no telpas kvadrātmetrāžas. Valsts pētījumu programma, lai palīdzētu mazināt vīrusa sekas, Valsts pētījumu programma “Covid-19 seku mazināšanai” programmas ietvaros īstenoja 10 projektus, no kuriem viens inženiertehnisko risinājumu jomā apstiprināts bija projekts ar nosaukumu “Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project)”[4]. Autora ieguldījums šajā projektā bija monitorēšanas daļai, par šo tēmu arī ir bakalaura darbs.

## 1.1. Problēmas apraksts

Problēma, ko monitorēšanas daļā nepieciešams risināt, ir cilvēku monitorēšana. Veicot novērošanu, jāievēro projekta dalībnieku privātums, kā arī izvēlētajam risinājumam jābūt lietotājam draudzīgam, precīzam un iegūtajiem datiem jābūt pārsūtāmiem uz projekta kopējo datubāzi.

Lai apzinātu esošo situāciju un izdarītu izglītotāku lēmumu nepieciešams apskatīt esošos risinājumus, izvēlēties visatbilstošāko un mācīties no jau izplatītām sabiedrības virzītām lietotnēm.

## 2. Esošu risinājumu izpēte

### 2.1. Citas sabiedrības virzītas sistēmas

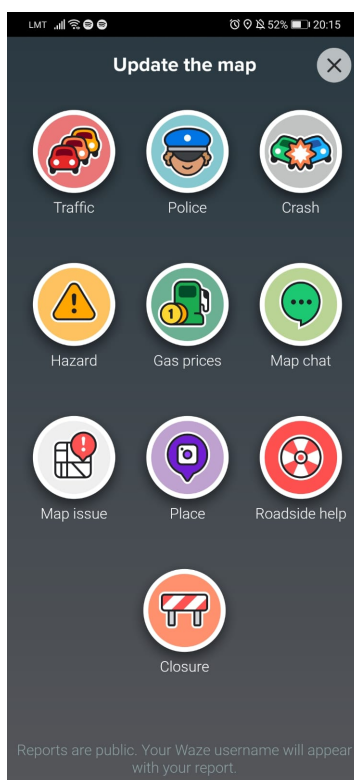
Lai labāk spētu izstrādāt lietotni, kas pielieto lietotāju datus, lai uzlabotu tās datu precizitāti, nepieciešams izpētīt kā dažādas citas, plaši izplatītas sabiedrības virzītas sistēmas strādā. Lai objektīvi spētu izpētīt esošās lietotnes nepieciešams noteikt specifiskas vadlīnijas, kuras izpētīt:

- Kādēļ izvēlēta lietotne ir sabiedrības virzīta;  
Nepieciešams noteikt precīzus iemeslus, kāpēc apskatītā lietotne ir sabiedrības virzīta.
- Lietotāja ieguldījums pret ieguvumu.

Nepieciešams novērtēt kādus datus lietotājs iegulda lietotnē un ko saņem no tās - kāpēc to lietot. Šī būs arī galvenā un svarīgākā vadlīnija kuru apskatīt.

#### 2.1.1. Waze

*Waze* ir GPS navigācijas mobilā lietotne. Tā nodrošina detalizētu navigācijas informāciju un lietotāja iesniegtos ceļojuma laikus un maršruta informāciju, vienlaikus lejupielādējot no vietas atkarīgu informāciju, izmantojot mobilo telefonu tīklu.



**Attēls 2.1.** *Waze* lietotnes iespējamo atzīmējumu skats.

Attēlā 2.1. redzami pāris iespējami varianti kā *Waze* lietotāji brīvprātīgi spēj uzlabot lietotnes stāvokli, piemēram, atzīmēt, ka uz ceļa apstājusies mašīna caur *Hazard* ikonu vai potenciālu policistu uz ceļa lietojot *Police* ikonu. Laigam šie ir brīvprātīgie varianti, kā lietotājs var uzlabot lietotnes ekosistēmu, bez lietotāju iniciatīva *Waze*, lietojot braucēju atrašanas vietu, rēķina vidējo satiksmes ātrumu konkrētā ceļa galumā, kas, var palīdzēt citiem braucējiem, kas vēl tikai plāno savu braucienu.

*Waze* darbības princips padara to par sabiedrības virzītu sistēmu. Lietotāja iespēja atzīmēt uz ceļa notiekošo un apziņot pārējos lietotājus par iespējamām problēmām to maršrutos ir galvenie lietotnes darbības principi.

*Waze* no lietotājiem ievāc lietotāja konta informāciju (lietotāja vārds, epasts), kā arī visu informāciju, ar kuru lietotājs pats izvēlas dalīties (lietotājavārds, ziņojumi, kas saistīti lietotājavārdu, telefona numurs, utt). Vēl *Waze* ievāc jebkādas komunikācijas datus, kurus lietotājs lietotnē ievada. [1]

Lietotājam, daloties ar šādiem datiem, tas pretī saņem pārredzamu un detalizētu maršrutu un reālā laika atjauninājumus par traucējumiem vai atbrīvojumiem uz ceļiem maršrutā.

### 2.1.2. TikTok

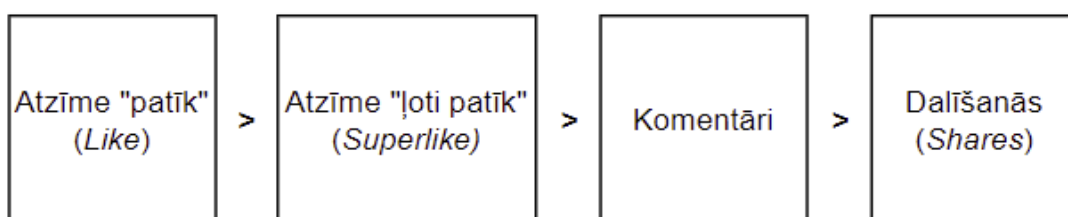
*TikTok* ir video koplietošanas sociālo tīklu pakalpojums. Sociālo mediju platformu izmanto, lai veidotu dažādus īsas formas videoklipus, kuru ilgums ir no 15 sekundēm līdz vienai minūtei.

Ķīnas kompānijas *ByteDance* lietotne regulāri no lietotāja ievāc visu, ko var par pašu ierīci (ierīces ID, ierīces tipu, ierīces zīmols, OS versiju, OS API, utt). Vēl lietotnes izstrādātāji no lietotāja ievāc datus par reģionu (laika zonu, esošo reģionu, lokalizāciju, utt)[2]. Var redzēt, ka lietotne no lietotāja regulāri neievāc neko neparastu. Taču problēmas ar datu ievākšanu sākas, kad lietotni sāk lietot tās pilnībā - visas ziņas, kuras lietotāji sūta tiek saglabātas. Protams tiek ievākti arī dati par video skatīšanos, lai labāk spētu noteikt video ieteikumus.

*TikTok* lietotāji no sniegtajiem datiem iegūst personalizētus video (no 10 rekomendētiem video, 8 balstās uz iepriekš redzētiem video un citu līdzīgi domājošu cilvēku

trendiem, 2 tiek rekomendēti balstoties uz atrašanās vietu). Vēl lietotāji no *TikTok* iegūst iespēju komunicēt viens ar otru platformas iekšienē.

Lietotājiem tiek dotas iespējas ietekmēt gan savu, gan citu platformu lietotāju ieteikšanas algoritmu, lietojot *TikTok* iebūvētos vērtēšanas rīkus (atzīme “patīk”, atzīme “ļoti patīk” (*superlike*), komentāri, dalīšanās (*shares*)). Katram no šiem interaktivitātes rīkiem ir savs svars, tas tiek parādīts attēlā 2.2. Šīs skatītāju/video veidotāju savstarpējās interaktivitātes dēļ *TikTok* ir labs un moderns piemērs, kā ļaut lietotājiem uzlabot savu un citu pieredzi.



**Attēls 2.2. *TikTok* vērtēšanas rīku svars**

Tika apskatīti daži labi piemēri sabiedrības virzītām lietotnēm, kuras izmanto lietotāju interaktivitāti, lai uzlabotu pašas lietotnes pakalpojumu. No šī var teikt, ka sabiedrības virzīta lietotne ir īstais variants, lai mazinātu neprecizitātes, vai lietotājiem piegādātu svarīgu informāciju.

## 2.2. Cilvēku monitorēšana telpā

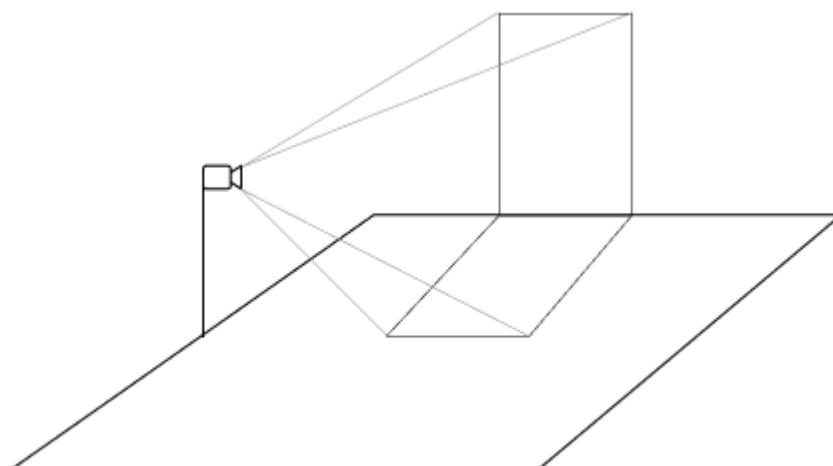
Cilvēku monitorēšanai nepieciešams apskatīt dažādus risinājumus, kas saglabātu dalībnieku privātumu un joprojām nodrošinātu datu precizitāti. Risinājuma ietvaros tika apskatīti arī citi monitorēšanas paņēmieni, taču darbā aplūkosim tikai vispotenciālākos variantus.

### 2.2.1. Digitālo attēlu analizēšana

Kā pirmo no potenciāliem cilvēku monitorēšanas variantiem darbā aplūkosim digitālo attēlu analizēšanu, lai iegūtu precīzu cilvēku skaitu no attēliem. Visi kameras kadri tiktu

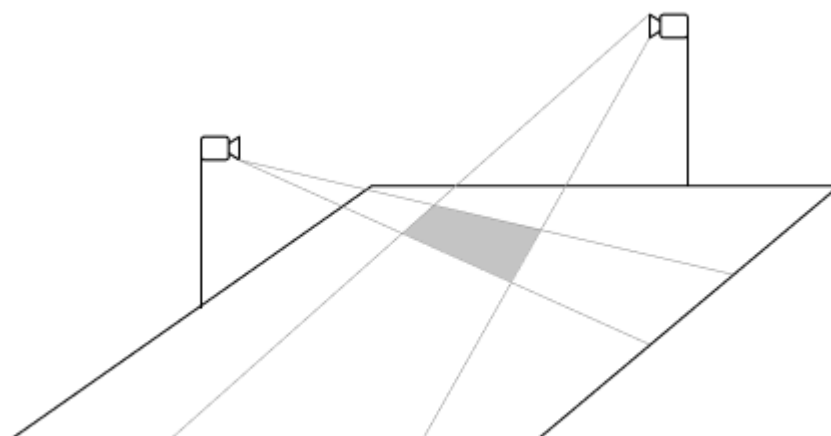
pārstrādāti un izskaitīti uz vietas, lai palielinātu ātrumu un mazinātu potenciālās savienojuma problēmas.

Iespējamais veids, kā šis varētu tikt realizēts ir ar vienu kameru, kas vērsta pret ieeju[9]. Neliela telpa pēc durvīm tiek uzskatīta par “skaitāmo zonu”, kuras šķērsošanas laikā attēls tiek uztverts, izanalizēts un cilvēks tiek pieskaitīts pie telpas kopējo cilvēku skaita. Šādā situācijā tiek saskaitīti visi ienākošie un atskaitīti visi aizejošie cilvēki. Šis variants atduras pret ideju, ka istabai ir vairākas ieejas, kurām katrai nepieciešama sava kamera.



**Attēls 2.3. optiskās kameras 1. risinājuma shēma**

Vēl iespējams realizēšanas variants optisko kameras risinājumam ir noklāt visu apstaigāto zonu ar kameru redzes lokiem, šādi izveidojot tīklu no dažādiem skatiem. Katram šādam skatam nepieciešams veikt analīzi.



**Attēls 2.4 optiskās kameras 2. risinājuma shēma**

Attēlā 2.4 var redzēt galveno problēmu šāda veida risinājumam - kameru redzamībai pārklājoties to nepieciešams reģistrēt. Parasti šāda veida cilvēku un objektu atpazīšanu veic ar neironu tīkliem[10], kurus trenē ar ļoti lielu datu kopu, lai tas spētu atpazīt cilvēkus no visām pusēm un augstumiem.

Risinājumam ar optiskām kamerām ļoti liels pluss ir, precizitāte, ko var iegūt specifiskos variantos, kas ir nepārspējama, taču tā nenāk bez trūkumiem - privātums tiek zaudēts, kad tiek pielietota jebkāda veida optiskā kamera. Augstāk minētā problēma ar kameru redzamības pāklāšanos, vai “*handoff problēma*” (nodošanas problēma), šīs problēmas risināšana dzīvā laikā prasa ļoti lielu skaitļošanas jaudu. Šī iemesla dēļ, optiskās kameras nav ieteicamais risinājums konkrētajai cilvēku monitorēšanas problēmai.

### 2.2.3. Bluetooth BLE bākas

Vēl tika apskatīts risinājums lietojot *Bluetooth* BLE bākas, lai nodefinētu konkrētu telpu un reģistrēt apkārtējos cilvēkus pie šīs bākas. Risinājumam tika apskatītas vairāki bāku ietvari, taču darbā sīkumos izpētīsim tikai dažas[12].

### 2.2.3.1 iBeacons

*Apple iBeacons* ir pirmā BLE bāka, kas iznāca, tādēļ pēcnācēju bākas ņem piemēru datu formātā tieši no *iBeacons* [11]. Tie lieto fiksētu reklamēšanas prefiksu: 0x0201061AFF004C0215. Šī heksadecimāla dati tiek dalīti zemāk, tabulā 2.1.

**Tabula 2.1 iBeacons Reklamēšanas prefiksa sadalījums.**

Lauks	Izmers	Vērtība	Apraksts
Reklamēšanas karodziņi ( <i>Adv Flags</i> )	3 biti	0x020106	Šis precizē, ka ierīce tikai pārraida, tā neuztver.
Reklamēšanas galvene ( <i>Adv Header</i> )	2 biti	0x1AFF	Sekojošie dati ir 26 bitus gara un ir Izstrādātāja Specifiski Dati.
Kompānijas ID	2 biti	0x004C	<i>Apple</i> kompānijas atpazīstamības kods
<i>iBeacon</i> tips	1 bits	0x02	Sekundārs ID, kas apzīmē bāku tuvumu, ko lieto visas <i>iBeacons</i> tipa bākas.
<i>iBeacon</i> garums	1 bits	0x15	Nosaka, ka atlikušais garums ir 21 biti (identificējošā informācija, 16+2+2+1).

**Tabula 2.2 iBeacons profila skeleta sadalījums.**

Lauks	Izmērs	Apraksts
UUID	16 biti	Unikāls ID, kas vienu <i>iBeacons</i> bāku atpazītu no citas.
<i>Major</i> numurs	2 biti	Numurs no 1 līdz 65,535, kas palīdz atpazīt bāku apakškopu lielākā grupā.
<i>Minor</i> numurs	2 biti	Tāpat kā <i>Major</i> numurs tikai lai atpazītu konkrētu bāku.
TX jaudas līmenis	1 bits	Šis numurs norāda signāla spēku vienu metru no ierīces.

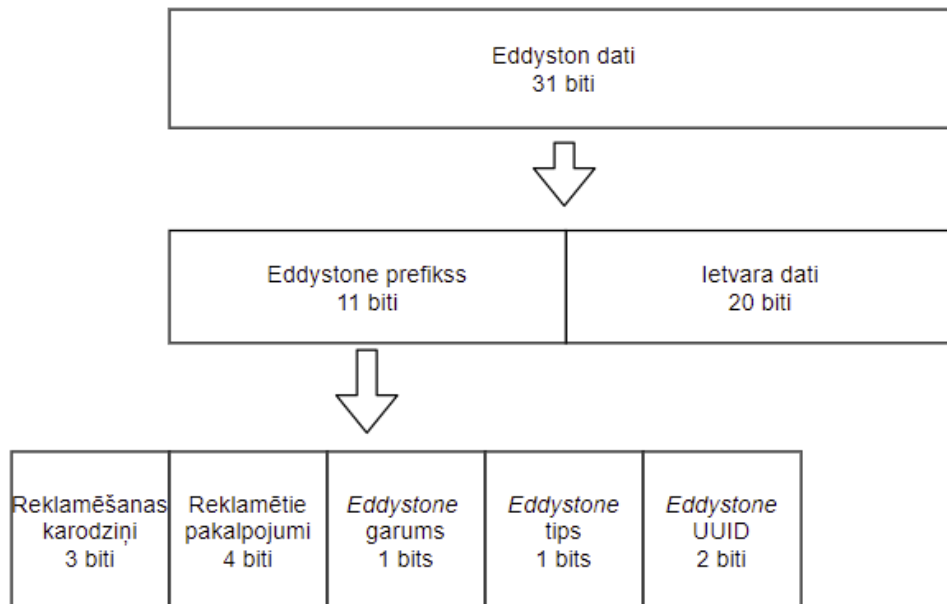
Tabulā 2.2 redzamas 4 svarīgākās *iBeacons* tipa bāku datu detaļas: UUID, *Major* numurs, *Minor* numurs, Tx jaudas līmenis.

UUID, *Major* numurs, *Minor* numurs sniedz identificējošu informāciju priekš bākas. Kad ierīce (parasti viedtālrunis) saņem šo informāciju un ar to atsaucas uz datubāzi. Pati bāka sevī nesatur nekādus aprakstošus datus, tam nepieciešama ārēja datubāze.

Tx jaudas līmeņa lauks ir mērītais signāla spēks, lai noskaidrotu cik tuvu gudrā ierīce ir no bākas. Šis var tikt parādīts kā aptuvenais novērtējums (netālu/tālu/ārpus uztveres zonas) vai arī kā precīzāki mērījumi metros.

### 2.2.3.2 Eddystone

*Eddystone* ir Google ieviests atvērts protokols. *Eddystone* bākas spēj izziņot datus 4 dažādos ietvara tipos: *Eddystone-UID*, *Eddystone-EID*, *Eddystone-URL* un *Eddystone-TLM*, kas visi strādā gan priekš *Android*, gan *iOS* operētājsistēmām[13].



**Attēls 2.5** *Eddystone* datu sadalījums pa bitiem

**Tabula 2.3. Eddystone reklamēšanas prefiksa sadalījums pa bitiem**

Lauks	Izmers	Vērtība	Apraksts
Reklamēšanas karodziņi ( <i>Adv Flags</i> )	3 biti	0x020106	Tāds pats, kāds <i>iBeacon</i> protokolam, tas norāda to, ka ierīce tikai pārraida un neuztver.
Reklamētie pakalpojumi	4 biti	0x0303AAFE	Atsaucas uz pilnīgo sarakstu ar 16-bitu <i>Service</i> UUID, kuram jāsaturs <i>Eddystone</i> pakalpojuma UUID (0xFEAA). Šis ir iekļauts, lai būtu iespējams skenēt fonā uz iOS ierīcēm.
<i>Eddystone</i> garums	1 bits	0x??	Atlikušās paketes garums. Šis var variēt no 6 bitiem līdz 20. Atšķiras balstoties uz ietvara datu garumu.
<i>Eddystone</i> tips	1 bits	0x16	Pakalpojuma datu tipa vērtība.
<i>Eddystone</i> UUID	2 biti	0xAAFE	<i>Eddystone</i> UUID.

Kā augstāk minēts, *Eddystone* ir 4 dažādi ietvari taču šī darba un risinājuma ietvaros detalizētāk apskatīsim tikai *Eddystone-UID*, jo tas ir visdrīzākais kandidāts cilvēku monitorēšanas risinājumam.

*Eddystone-UID* ietvara specifikācija ir ļoti līdzīga *iBeacon*. Līdzīgi kā *iBeacon* protokolā, *Eddystone-UID* ļauj ierīcei, lietojot tās ID, iesākt vēlamu darbību, piemēram, pieprasīt datus no servera. Tā lieto 16 bitus no ietvara, lai pieglabātu bākas ID, tas ir 10 bitu nosaukumvieta, kuru izmanto lai grupētu konkrētu bāku setu (piemēram organizāciju) un 6 bitu instanci, ko izmanto lai identificētu individuālas bākas.

Ietvara tips 1 bits	Diapazona dati 1 bits	Nosaukumvieta ID 10 biti	Instances ID 6 biti	RFU 2 biti
------------------------	--------------------------	-----------------------------	------------------------	---------------

**Attēls 2.6. Eddystone-UID ietvara datu sadalījums pa bitiem**

Diapazona dati ir pārraides spēks decibel-milivatos (dBm), kuru pārraida *Eddystone* bāka 0 metru attālumā. Šis ir savādāk nekā *Apple iBeacon* protokola - pārraides spēks tiek mērīts no viena metra. Taču *Google* rekomendē, ka labākais veids, kā noteikt precīzo vērtību, kuru šajā laikā ievadīt ir nosakot *Eddystone* bākas izvadi no 1 metra attāluma un pieskaitot 41 dBm iegūtajai vērtībai. 41 dBm ir signāla zaudējums, kas notiek katru 1 metru.

**Tabula 2.4 *Eddystone-UID* ietvara datu sadalījums pa bitiem ar aprakstiem.**

Lauks	Izmērs	Apraksts
Ietvara tips	1 bits	Izmanto, lai noteiktu <i>Eddystone</i> ietvara tipu. Vērtība šim UID tipam ir 0x00.
Diapazona dati	1 bits	Šis numurs norāda signāla spēku 0 metru attālumā.
Nosaukumvietas ID	10 biti	Unikāls ID, kuru izmanto, lai identificētu grupu ar bākām.
Instances ID	6 biti	ID, kas ļauj identificēt individuālu bāku.
RFU	2 biti	Rezervēts izmantošanai nākotnē, obligāti jābūt 0x0000.

Realizācijas pielietojot BLE bākas var būt dažādas, taču visiem nepieciešama gan kāda ārēja ierīce, kas veic skaitīšanu, gan arī ierīce no lietotāja puses, kas spēj uztvert bākas signālu un no iegūtā signāla veikt attiecīgos aprēķinus (atņemt vai pieskaitīt). Viens no variantiem šādai ierīcei ir mobilais telefons, jo mūsdienu telefoni ir aprīkoti ar dažādiem sensoriem, raidītājiem un uztvērējiem kurus var izmantot tieši šīs problēmas risināšanai. Šis ir arī ideāls variants, jo mūsdienu cilvēkam vienmēr pie rokas ir šāds mazs dators.

Kā jau iepriekš minēts - plusi šim risinājumam ir precizitāte, ko iegūst katram lietotājam pašam reģistrējoties pie bākas. Taču turpat ir arī trūkums, kas parādas no lietotāja sagaidot, ka tie izmantos un aktīvi piedalīsies, jo bieži būs cilvēki, kas nevēlēsies papildus lietas darīt, tādēļ šādam risinājumam dot tā saucamo burkānu, lai tiem būtu lielāka iniciatīva izmantot risinājumu un būt daļai no sistēmas.

## 2.2.4. CO2 mērījumu veikšana

Kā pēdējo cilvēku monitorēšanas iespēju darbā apskatīsim CO2 mērījumu interpretēšanu. Lai veiktu precīzus mērījumus, pielietojot CO2 mērīšanu, nepieciešams zināt telpas tilpumu un ventilācijas daudzumu.

$$C_i = C_0 + (C_s - C_0)e^{-\frac{Q}{V}(t-s)} + \left(1 - e^{-\frac{Q}{V}(t-s)}\right) \frac{M}{Q}$$

$$C_i = C_0 + (C_s - C_0)e^{-\frac{Q}{V}(t-s)} + \left(1 - e^{-\frac{Q}{V}(t-s)}\right) \frac{kn}{Q}$$

$$n = \frac{Q}{k \left(1 - e^{-\frac{Q}{V}(t-s)}\right)} \left(C_i - C_0 - (C_s - C_0)e^{-\frac{Q}{V}(t-s)}\right)$$

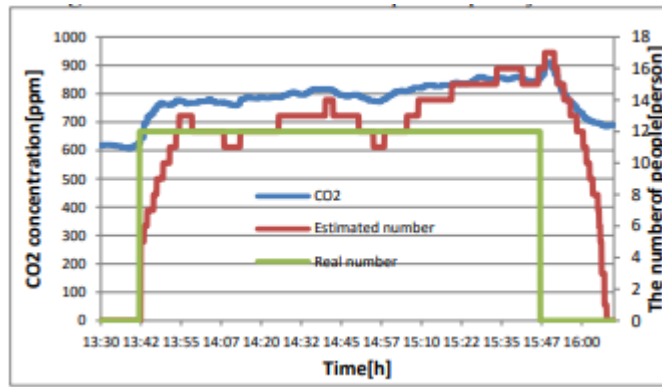
**Attēls 2.7. Cilvēku skaita noteikšana no CO2 ar pārveidojumiem.[14]**

Lietojot attēlā 2.7. redzamās formulas pārveidojumu, ir iespējams novērtēt cilvēku daudzumu, kas atrodas telpā. Attēlā 2.5. var redzēt visu formulas parametru atšifrējumus un kā tie tiktu noteikti.

Symbol	Name	Method
C[ppm]	Concentration of indoor pollutants	Measurement
C0[ppm]	Concentration of no pollutant sources	Estimation
M[m <sup>3</sup> /h]	Amount of pollutants generated	k · n
V[m <sup>3</sup> ]	Volume of the room	Measurement
Q[m <sup>3</sup> /h]	Amount of ventilation	Estimation
i,s[h]	Time	Measurement
k[m <sup>3</sup> /h · person]	CO2 emissions per person	Reference[6]
n[person]	Number of people	Estimation

**Attēls 2.8. Formulas atšifrējumu tabula [14]**

Pētījums, kurā tika izmantota šī formula veica testus mainot tikai ventilācijas daudzumu un cilvēku daudzumu. Rezultātā tika iegūti grafiki, kuros skaidri redzams, ka pielietojot šādu cilvēku mērīšanas variantu ir ļoti liela iespēja saskarties ar neprecizitātēm.



Attēls 2.9. Cilvēku daudzuma prognozes ar 957.3 m<sup>3</sup>/h ventilāciju [14]

### 3. Izvēlētās tehnoloģijas

Risinājuma realizācijā nepieciešams pielietot tehnoloģijas, kas risinās 1. nodaļā aprakstīto problēmu, izmantos piemērus no 2. nodaļā apskatītajām lietotnēm un izvēlētos cilvēku skaitīšanas veidus. Risinājuma realizācijai tiek izmantotas Bluetooth BLE bākas, ar kuru palīdzību lietotāji tiek reģistrēti telpām, Firebase[5] mākoņserviss, kas kalpo kā sistēmas serveris - tas reģistrē un autentificē lietotājus, uzkrāj datus par telpu aizņemtību, un ievāktos datus par telpu aizņemtību nosūta uz ārēju Mosquitto brokeri[7], lietojot MQTT protokolu. Sistēmas centrā tiek lietota Android mobilā lietotne, jo mūsdienu viedtālrunis ir pietiekami attīstīts un izplatīts, lai varētu piedalīties maksimāli daudz dalībnieku. Konkrētajā sistēmā dati par telpas gaisa kvalitāti tiek iegūti no ārēja, Jura Veldres izstrādāta API, kas pēc pieprasījuma tos piegādā konkrētajam, reģistrētajam lietotājam.



Attēls 3.1. Realizētās sistēmas shēma.

#### 3.1. Mobilais telefons

Ierīce, kas no lietotāja puses nodrošinās to dalību sistēmā jau tiek lietota ikdienišķi - mobilais telefons. Mūsdienu viedtālrunim ir pieejami visi nepieciešamie sensori un interneta savienojums, kas tiem ļauj ievākt, saņemt un nosūtīt nepieciešamos datus.

Mobilo lietotni priekš viedtālruna oriģināli bija plānots veidot izmantojot *React Native*[15], taču saskaroties ar problēmām *Bluetooth* bāku uztveršanā lietotne tika pārcelta uz *Android Native* ietvaru. *Native* lietotne tika rakstīta valodā *Kotlin*, nevis *Java*, jo autoram tā ir pazīstamāka valoda, kā arī *Kotlin* ļauj lietotnes kodu rakstīt daudz tīrāk un salīdzinoši ar *Javu*, daudz īsāk[16].

## 3.2. Firebase

*Firebase* ir *Google* izstrādāts mākoņserviss, kura galvenā funkcija ir palīdzēt mobilām un WEB lietotnēm nodrošinot tās ar viegli uztveramu un lietojamu *back-end* servisu. Galvenā problēma lietojot *Firebase* ir tā minimālās iespējas veikt sarežģītus pieprasījumus, jo viss, kas tiek uzglabāts ir viens JSON fails.

*Firebase* serviss piedāvā daudz dažādas iespējas, kas atvieglo gan izstrādi gan lietotāju pieredzi. Visus *Firebase* piedāvātos pakalpojumus risinājumā nevajadzēs izmantot, taču risinājumā tika pielietoti trīs piedāvātās funkcionalitātes:

- *Firebase* autentifikācija (*Firebase Authentication*)
- *Firebase* tiešsaistes datubāze (*Firebase Realtime database*)
- *Firebase* mākoņa funkcijas (*Firebase Cloud functions*)

### 3.2.1. Firebase autentifikācija

*Firebase* autentifikācija nodrošina *back-end* pakalpojumus, ērti lietojamus SDK un gatavas lietotāju saskarnes bibliotēkas, lai autentificētu lietotājus lietotnē. Tā atbalsta autentifikāciju, izmantojot paroles, tālruņa numurus, populārus identitātes nodrošinātājus, piemēram, *Google*, *Facebook*, *Twitter*, u.c. Šajā risinājumā tika aktivizēta tikai paroles un epasta autentifikācijas iespēja. *Firebase* autentifikācijai iespējams izveidot epastu verificēšanas rīku, taču tas netika pielietots, jo lietotne netika izplatīta.

### 3.2.2. Firebase tiešsaistes datubāze

*Firebase* tiešsaistes datubāze ir mākonī mitināta datubāze. Dati tiek glabāti kā JSON objekti un sinhronizēti ar katru pievienoto klientu. Veidojot starpplatformu lietotnes, izmantojot piedāvātos iOS, Android vai Javascript SDK, visi klienti koplieto vienu tiešsaistes datubāzes instanci un automātiski saņem atjauninājumus ar jaunākajiem datiem. Datubāzi iespējams arī nodrošināt ar iekšējiem noteikumiem, kas pēc noklusējuma ļauj visiem rakstīt un lasīt visus datubāzes datus.

Risinājumam tika izvēlēts lietot tiešsaistes datubāzi nevis *Firestore* datubāzi, glabājamo datu izmēra un maiņu biežuma dēļ. Dati, kurus nepieciešams glabāt ir ļoti mazi

(tikai kurās telpās un cik cilvēki), tādēļ izvēlēts lietot vienkāršākās struktūras datubāzi, protams arī datu izmaiņas sagaidāmas salīdzinoši bieži, tāpēc ir svarīgi, lai datubāzi nepārslogotu ar atkārtotām kolekciju un vērtību pāru izmaiņām. Risinājumā arī datubāzes noteikumi tika izmainīti, lai lietotājam dotu tikai tik daudz piekļuvi, cik nepieciešams.

### 3.2.3. Firebase mākoņfunkcijas

*Firebase* mākoņfunkcijas ir bez servera ietvars, kas ļauj automātiski palaist *back-end* kodu, reaģējot uz *Firebase* funkciju un HTTPS pieprasījumu izraisītiem notikumiem. Javascript vai TypeScript kods tiek glabāts *Google* mākonī un darbojas pārvaldītā vidē. Šādā veidā nav nepieciešams pārvaldīt un mērogot savus serverus.

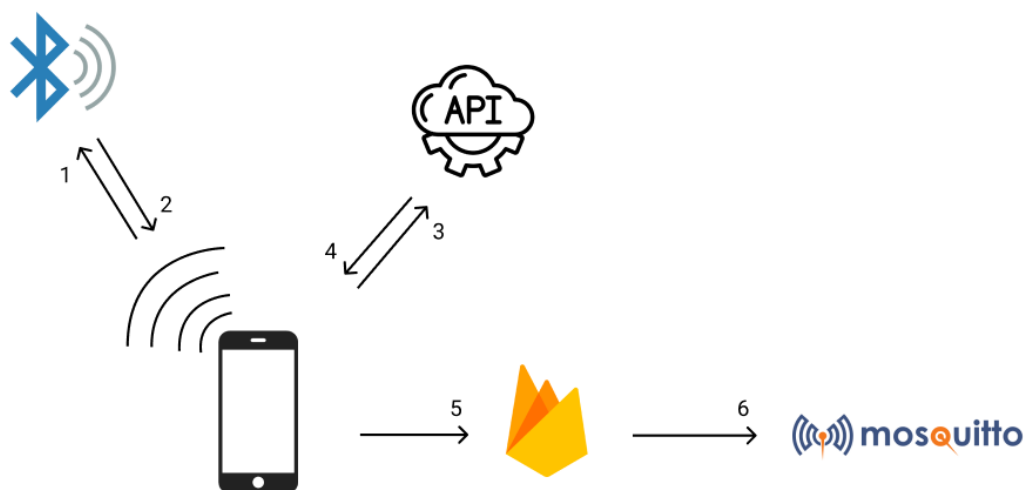
Risinājumā tika pielietotas trīs *Firebase* funkciju pieprasījumu izraisītu notikumu mākoņfunkcijas, kas rakstītas JavaScript.

### 3.3. *iBeacons* protokola BLE bāka

Kā telpas sensoru risinājums tika izvēlēts *iBeacons* nevis *Eddystone*, iespējamo datu pārraides dēļ. Ar *iBeacons* protokola lietošanu iespējams bākas dalīt izmantojot *Major* un *Minor* numurus, tādējādi, piemēram, atdalot stāvu kādā iestādē vai arī pašas iestādes.

## 4. Realizētais risinājums

### 4.1. Tehniskie risinājumi



Attēls 4.1. Realizētās sistēmas shēma.

Attēlā 4.1. ir redzama realizētā cilvēku monitorēšanas sistēmas shēma. Shēmā var redzēt, ka mobilā lietotne ir komunikācijas centrā.

1. Mobilā ierīce skenē apkārtesošo iBeacons tipa *Bluetooth* BLE bāku signālus;
2. iBeacons tipa *Bluetooth* BLE bāka pārraida savu UUID;
3. Pēc bākas uztveršanas, tiek sūtīts pieprasījums uz API;
4. API pārbauda saņemto bākas UUID. Ja tas ir reģistrēts, tad uz pieprasījumu atbild ar datiem par telpas gaisa kvalitāti. Ja tas nav reģistrēts, API datus nesūta;
5. Mobilā lietotne ar katru tuvākās bākas izmaiņu nosūta atjaunotos datus uz *Firebase* mākoņservisu;
6. Mākoņserviss, saņemot atjauninātus datus, tos nosūta uz ārēju datubāzi, lietojot MQTT protokolu.

#### 4.1.1. Firebase (Back-end)

##### 4.1.1.1. Firebase autentifikācija

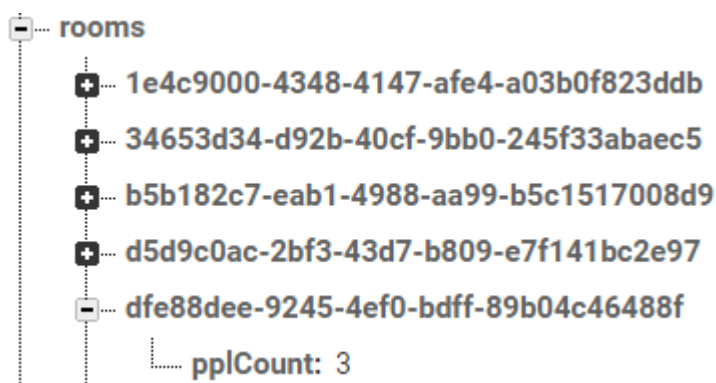
Firestore autentifikācija nepieciešama, lai veidotu drošu savienojumu un lietotājus nodrošinātu ar personalizētu ID, kuru vēlāk var izmantot datubāzē, lai veiktu ierakstus, kuriem nav iespējams atkārtoties, jo šie ID tiek veidoti unikāli.

#### 4.1.1.2. Firebase tiešsaistes datubāze



**Attēls 4.2. Lietotāju mezgls**

Attēlā 4.2. var redzēt lietotāju izkārtojumu datubāzē. Katram lietotāja mezglam ir piesaistīts attiecīgās telpas unikāls ID, kurā lietotājs pašreiz atrodas. Šis ID mainās, kad lietotājs ienāk citas bākas apkārtnē. Konkrētajā piemērā redzams, ka vienā telpā atrodas trīs dažādi lietotāji.



**Attēls 4.3. Bāku mezgls**

Attēlā 4.3. var redzēt bāku, jeb istabu izkārtojumu datubāzē. Katram telpas mezglam ir piesaistīts cilvēku skaits, attēlojot cik cilvēki pašreiz atrodas telpā. Šis skaits mainās, kad lietotājs ienāk vai iziet citas bākas apkārtnē. Konkrētajā piemērā redzams, ka vienā telpā atrodas trīs dažādi lietotāji.

```

1  {
2    "rules": {
3      ".read": false,
4      ".write": false,
5      "users": {
6        "$uid": {
7          ".write": "$uid === auth.uid"
8        }
9      },
10   "rooms": {
11     "$roomId": {
12       "reports": {
13         "$uid": {
14           ".write": "$uid === auth.uid"
15         }
16       }
17     }
18   }
19 }
20 }

```

Attēls 4.4. *Firestore* tiesšsaistes datubāzes noteikumi

Attēlā 4.4. redzami datubāzes noteikumi. Noteikumi domāti, lai mazinātu datubāzes piekļuvi galalietotājiem, kas šajā gadījumā ir visi mobilās lietotnes īpašnieki. Konkrētie noteikumi sarakstīti tā, lai nodrošinātu, ka lietotāji spēj ierakstīt (gan mainīt, gan jaunu ierakstīt) tikai zem sev piešķirta ID mezgla.

#### 4.1.1.3. *Firestore* mākoņfunkcijas

```

exports.updateRooms = functions
  .database
  .ref('/users/{userID}')
  .onWrite((change, context) => {
const newRoomID = change.after.val().roomId
const oldRoomID = change.before.val().roomId

const removePersonRef = db.ref(`rooms/${oldRoomID}`).child('pplCount')
const addPersonRef = db.ref(`rooms/${newRoomID}`).child('pplCount')

const removeUserReport = removeReport(context.params.userID, oldRoomID)
const removePerson = removePersonRef.transaction(count => {
  return count - 1
})

const addPerson = addPersonRef.transaction(count => {
  return count + 1
})

return Promise.all([addPerson, removePerson, removeUserReport])
});

```

Attēls 4.5. *Firestore* mākoņfunkcija reģistrētu lietotāju skaitīšanai

Attēlā 4.5. redzama pirmā *Firebase* mākoņfunkcija, kas, reaģējot uz lietotāja mezgla izmaiņām, atskaita reģistrētu lietotāju no iepriekšējās telpas un pieskaita to pie jaunās telpas, atpakaļ atgriežot solījumu (Promise), kas mākoņfunkcijām ļauj atrisināt datubāzes darbības.

```
exports.parseCountReports = functions
  .database
  .ref('/rooms/{roomId}/reports/{userID}')
  .onWrite((change, context) => {
const oldReport = change.before.val() !== null ? change.before.val().count : 0
const roomId = context.params.roomID
const uid = context.params.userID

const reportRef = db.ref(`rooms/${roomId}/reports`)
const pplCountRef = db.ref(`rooms/${roomId}`).child('pplCount')

var reportCount = 0
var pplCountSum = 0
var previousSum = 0
var previousCount = 0

return reportRef.once('value', (snapshot) => {
  if (snapshot.exists()) {
    for (const [key, value] of Object.entries(snapshot.val())) {
      if (key !== uid) {
        previousSum += value.count
      }
      reportCount++
      previousCount++
      pplCountSum += value.count
    }
    previousSum = oldReport ? previousSum + oldReport : previousSum
  } else {
    previousSum = oldReport ? oldReport : previousSum
    previousCount = 1
    pplCountSum = 0
    reportCount = 1
  }
}).then(() => {
  return pplCountRef.transaction((count) => {
    const oldAvg = Math.ceil(previousSum / previousCount)
    count = count - oldAvg
    const newAvg = Math.ceil(pplCountSum / reportCount)
    count += newAvg
    return count
  })
})
})
```

Attēls 4.6. *Firebase* mākoņfunkcija neregistrētu lietotāju skaitīšanai

Attēlā 4.6. redzama mākoņfunkcija, kas reaģē uz lietotāju nosūtītiem ziņojumiem par telpā esošo cilvēku skaitu. Funkcija pārbauda vai lietotājam jau ir bijis ziņojums, ja ir bijis, to pārraksta pirmstam izrēķinot vidējo no jau esošā cilvēku skaita.

```

exports.pplCountChange = functions
  .database
  .ref('/rooms/{roomId}/pplCount')
  .onWrite(async (change, context) => {
    const roomId = context.params.roomID
    if (roomId !== "null") {
      const newPplCount = change.after.val()
      const mqtt = await publishViaMqtt(roomID, newPplCount)
      return mqtt
    }
  })

const publishViaMqtt = async (roomId, peopleCount) => {
  const topic = 'VPP-APP/beacons'
  var message = {
    people_count: peopleCount,
    time: new Date(),
    beacon_id: roomId
  }
  message = JSON.stringify(message)
  return new Promise((resolve, reject) => {

    client.publish(topic, message, (err) => {
      if (err) {
        console.log("MqttError publish:" + err.message);
        reject(err);
      } else {
        console.log("Message published")
        resolve()
      }
    })
  })
}

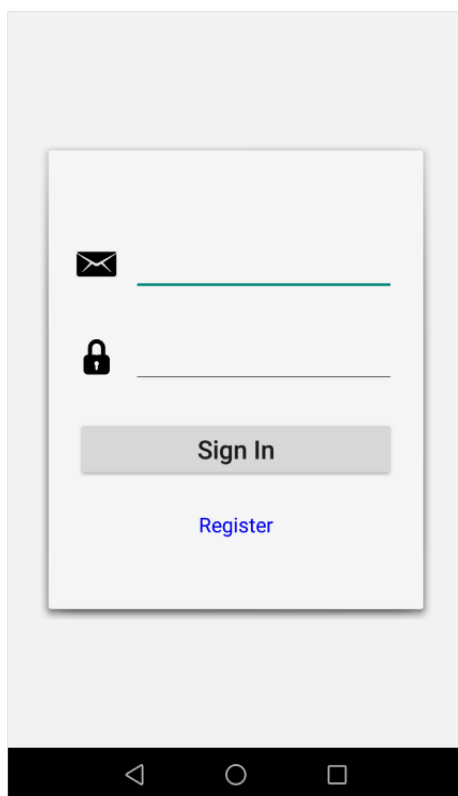
```

**Attēls 4.7. *Firestore* mākoņfunkcija datu pārsūtīšanai pār MQTT**

Attēlā 4.7. redzama mākoņfunkcija, kas reaģē uz cilvēku skaita maiņām. Funkcija datus pārtaisa JSON objektā un tos nosūta uz pirmstam izveidotu savienojumu, lietojot MQTT protokolu [17].

## 4.1.2. Android mobila lietotne (Front-end)

### 4.1.2.1. Ekrānskatī un dizains

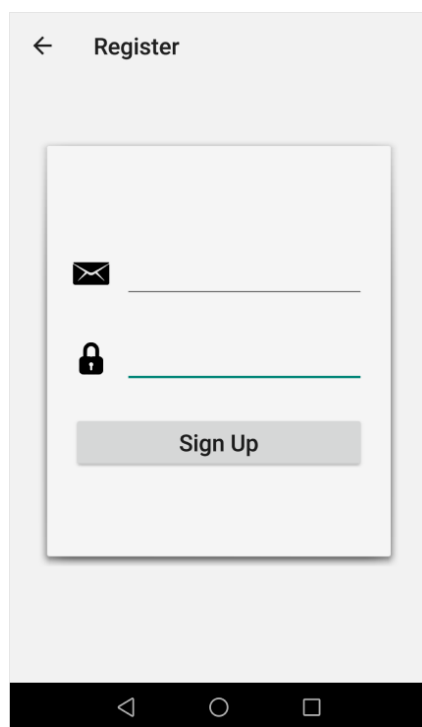


**Attēls 4.8. Autorizācijas skats.**

Autorizācijas skatā iespējams ievadīt epasta adresi, paroli un autorizēties sistēmā.

Tabula 4.1. Autorizācijas skata paziņojumu saraksts.

Paziņojuma numurs	Teksta apraksts	Sagaidāmais paziņojums
1	Lietotājs ievada pareizu epastu, bet nepareizu paroli	“Password is invalid or the user does not have a password.”
2	Lietotājs ievada nepareizi formatētu epastu	“The email address is badly formatted.”
3	Lietotājs ievada nepareizu epastu un paroli	“There is no user record corresponding to this identifier. The user may have been deleted.”
4	Lietotājs neievada neko vai epastu, vai paroli	“Please enter password and/or email.”

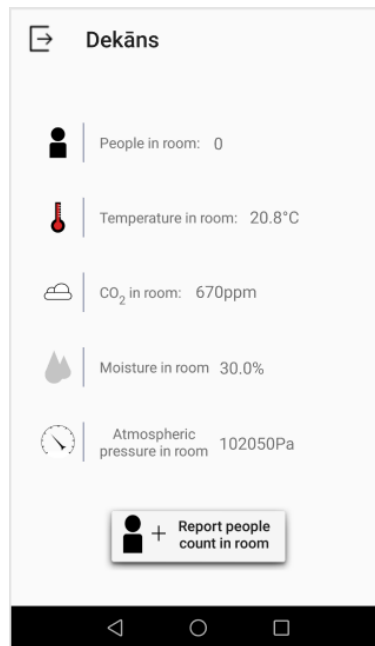


**Attēls 4.9. Reģistrācijas skats.**

Reģistrācijas skatā iespējams ievadīt epasta adresi, paroli un reģistrēties sistēmā.

Tabula 4.2. Reģistrācijas skata paziņojumu saraksts.

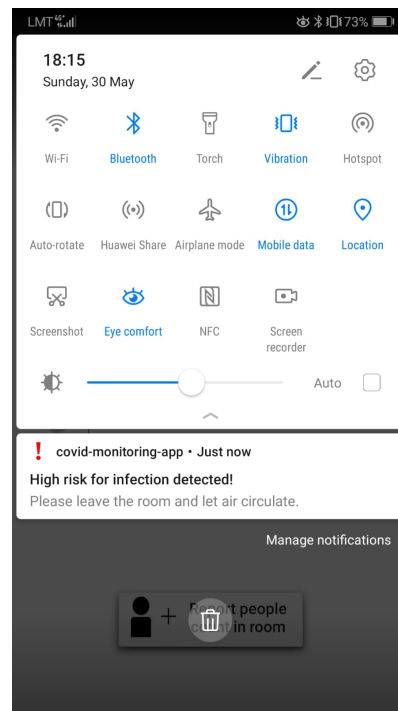
Paziņojuma numurs	Teksta apraksts	Sagaidāmais paziņojums
5	Lietotājs ievada pareizu epastu, bet nepareizu paroli	“Password is invalid or the user does not have a password.”
6	Lietotājs ievada nepareizi formatētu epastu	“The email address is badly formatted.”
7	Lietotājs ievada nepareizu epastu un paroli	“There is no user record corresponding to this identifier. The user may have been deleted.”
8	Lietotājs neievada neko vai epastu, vai paroli	“Please enter password and/or email.”
9	Lietotājs ievada paroli, kas īsāka par 6 simboliem	“The given password is invalid. [Password should be at least 6 characters]”



**Attēls 4.10. Telpas vides kvalitātes skats dekāna telpai.**

Telpas vides kvalitātes skatā iespējams novērot īsti laika izmaiņas telpas gaisa un vides kvalitātē. Lietotājam tiek arī dota iespēja zinot par cilvēku skaitu telpā.

Ja gaisa kvalitāte vai cilvēku skaits telpā pārsniedz ieteicamo, tad lietotāji, kas atrodas konkrētajā telpā saņems ziņu sava viedtālruna paziņojumu panelī, kā tas redzams attēlā 2.11.



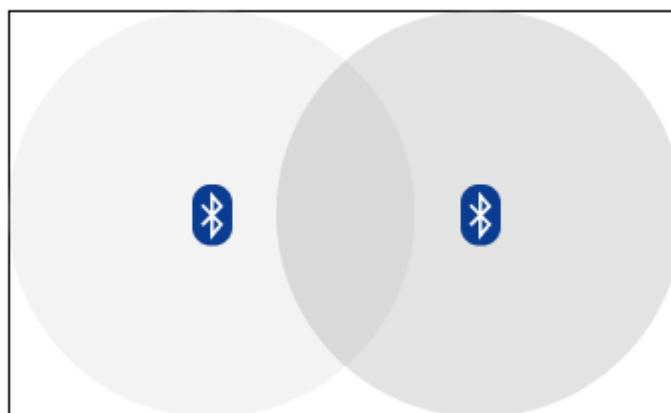
**Attēls 4.11. Lietotāja paziņojumu panelis, pēc kvalitātes sliekšņa pārsniegšanas.**

#### 4.1.2.2. Strādāšanas principi

Lietotājam autorizējoties sistēmā tam tiks prasīts, lai tas ieslēdz gan ierīces *Bluetooth*, gan GPS (šie ir obligāti, lai skenētu apkārtni, meklējot bākas). Tālāk ienākot bākas pārraides zonā, lietotājs telpas vides kvalitātes skatā varēs apskatīt cik cilvēki pašlaik atrodas telpā un citus gaisa kvalitātes mērījumus. Lietotājam tiek dota izvēle cilvēku skaita mērījumu ietekmēt lietojot ziņošanas pogu. Kad telpas gaisa kvalitāte vai cilvēku daudzums pārsniedz ieteicamo, visi lietotāji, kuri ir savienoti ar konkrētās telpas bākām, saņems paziņojumu. Pēc telpas un bākas apziņošanas zonas pamešanas, lietotāji ar ieslēgtām ierīces *Bluetooth* un GPS tiks atvienoti no telpas.

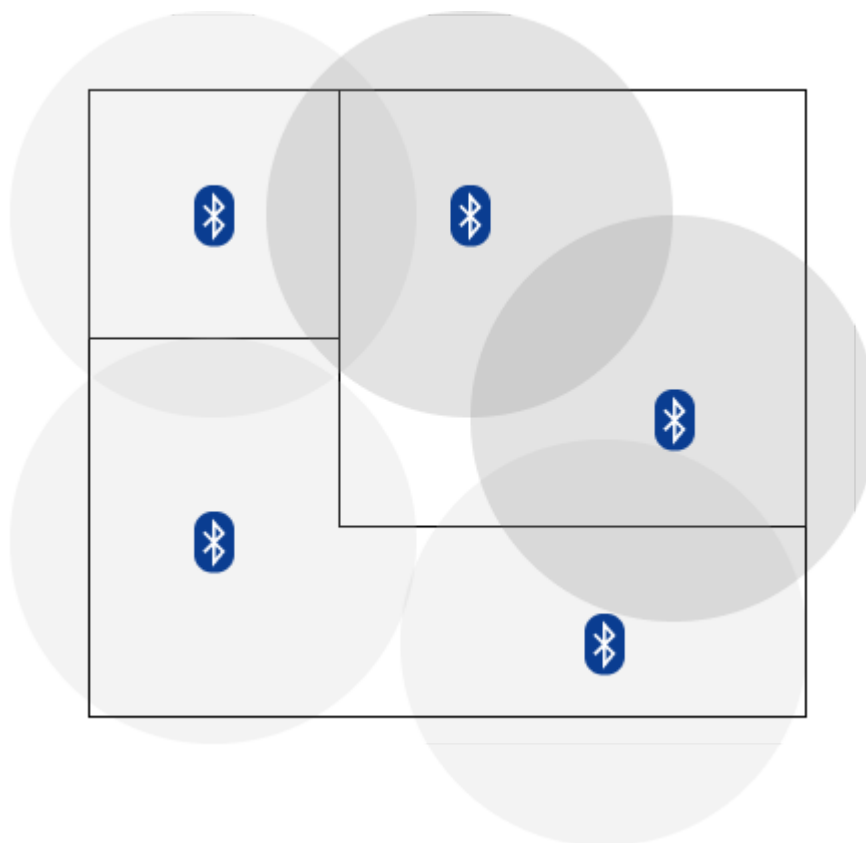
#### 4.1.3. Bluetooth BLE bākas (Sensori)

Ņemot vērā, ka bākām ir noteikts apziņošanas attālums, tās tika izvietotas telpā, tā, lai tās pārklātu maksimāli daudz telpas, izmantojot optimālu skaitu ar raidītājiem.



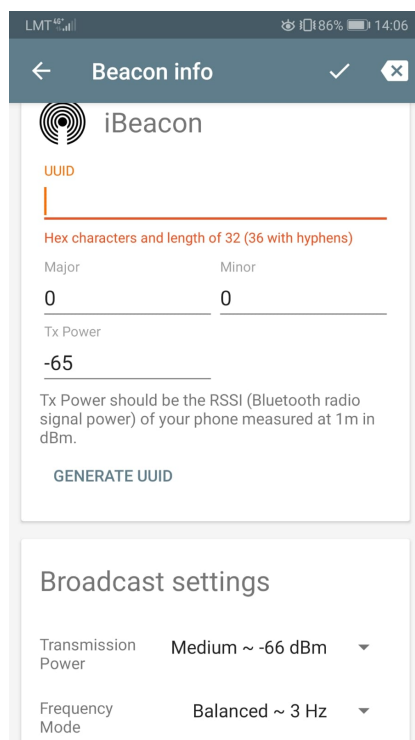
**Attēls 4.12. piemērs vienas istabas *Bluetooth* BLE izvietojumam.**

Attēlā 4.12. redzams potenciāls variants bāku izvietojumam vienas istabas izklājumā. Šādā situācijā iespējams arī ievietot bākas katrā no istabas stūriem, tādējādi nodrošinot maksimāli lielu pārklājumu un minimālu iespēju izveidot nepareizus savienojumus starp telpām, taču, lai izmantotu optimālu skaitu, pietiek ar divām bākām uz šāda izmēra telpu.



**Attēls 4.13. piemērs vairāku istabu *Bluetooth* BLE izvietojumam.**

Attēlā 4.13. redzams potenciāls piemērs bāku izvietojumam sarežģītākā telpu izkārtojumā. Šāda veida izkārtojumā nepieciešams ievērot visu bāku izkārtojumu, ne tikai vienas telpas, jo iespēja lietotājiem savienoties ar citu telpu bākām ir paaugstināta. Lai risinātu šo problēmu bāku novietojumiem jābūt izklātiem, tā, lai jebkurā telpas punktā tuvākā bāka atrodas tajā pašā telpā.



**Attēls 4.14. lietotnes *Beacons Simulator* bākas izveides skats**

Attēlā 4.13. redzams galvenais bākas veids, kas tika izmantots risinājumā - simulator. Izvēle lietot simulatoru tika izdarīta, jo simulator ir daudz pieejamāks, un to ir ātrāk pieregulēt attiecīgām vajadzībām. Simulātorā, kā parastai *iBeacons* bākai, var piešķirt UUID, *Major* numuru un *Minor* numuru, kā arī Tx spēku.

## 5. Rezultāti

### 5.1. Testēšana

Risinājuma realizācijas beigās tika veikta manuālā testēšana. Katra risinājuma daļa tika testēta atsevišķi, taču *Bluetooth* bākas tika testētas lietojot gan speciālu mobilo lietotni, kas tikai norādīja bāku attālumu no mobilās ierīces, gan gala produkta lietotni.

**Tabula 5.1. Mobilās lietotnes testi**

Testa identifikators	Testa teksta apraksts	Sagaidāmais rezultāts	Rezultāts
ML1	Viens mobilais telefons, ar ieslēgtu bāku simulatoru tika novietots 5 metru attālumā, otrs 1 metra attālumā. Tika ieslēgta bāku skenēšanas lietotne.	Bāku skenēšanas lietotnē bāku attālumi ir pareizi attēloti.	OK
ML2	Lietotājs mēģina sistēmai pierēģistrēties ar pareizi formatētu epasta adresi un drošu paroli.	Lietotājs tiek pierēģistrēts sistēmā un tiek nogādāts uz galveno datu pārskatu.	OK
ML3	Lietotājs mēģina sistēmai pieslēgties ar reģistrētu epasta adresi un paroli.	Lietotājs tiek savienots ar sistēmu un tiek nogādāts uz galveno datu pārskatu.	OK
ML4	Lietotājs vēloties iziet no lietotnes nospiež izrakstīšanās pogu.	Lietotājs tiek atvienots no sistēmas, visi tā dati par telpā esošajiem cilvēkiem tiek noņemti un lietotājs tiek nogādāts autorizācijas skatā.	OK
ML5	Lietotājs atrodies bākas teritorijā ieslēdz mobilās lietotnes <i>Bluetooth</i> un GPS iespējas.	Lietotājs lietotnē saņem datus par telpas gaisa kvalitāti.	OK
ML6	Lietotājs pārvietojas no vienas bākas klātbūtnes uz otru ar ieslēgtām mobilās lietotnes <i>Bluetooth</i> un GPS iespējām.	Lietotājs lietotnē saņem datus par telpas gaisa kvalitāti.	OK
ML7	Lietotājs pēc “Report people count in room” pogas nospiešanas	Lietotne datus saņemot no datubāzes atjauno telpas	OK

	ievada esošo cilvēku skaitu telpā.	skatā redzamos datus par telpā esošajiem cilvēkiem.	
ML8	Lietotājam atrodies telpā, izmainās gaisa kvalitātes mērījumi.	Lietotne atjauno redzamos datus.	OK

**Tabula 5.2. *Firestore* testi**

Testa identifikators	Testa teksta apraksts	Sagaidāmais rezultāts	Rezultāts
FB1	Mobilās lietotnes tuvāko bāku nosūta uz datubāzi.	<i>Firestore</i> funkcijas reģējot uz izmaiņām datos palielina konkrētās telpas cilvēku skaitu.	OK
FB2	Cilvēku skaits kādā no telpām palielinās.	<i>Firestore</i> funkcijas uztver izmaiņas un izmainīto cilvēku skaitu nosūta uz ārēju MQTT brokeri.	OK
FB3	Lietotājs pēc “Report people count in room” pogas nospiešanas ievada esošo cilvēku skaitu telpā.	<i>Firestore</i> funkcijas saņemtos datus apstrādā un pievieno telpā esošo cilvēku skaitam	OK

## Secinājumi

Darba izstrādes laikā autors iepazinās ar esošu, labi izstrādātu sabiedrības virzītu mobilo lietotņu piemēriem, dažādām cilvēku monitorēšanas pieejām, kā arī padziļinātāk izpētīja izmantotās tehnoloģijas un risinājumu variantus. Pēc izpētes tika realizēts izplānotais risinājums pielietojot *Firebase* mākoņservisu, *Android* mobilo viedtālruni, kuram lietotne tika rakstīta *Kotlin* programmēšanas valodā. Kā cilvēku monitorēšanas rīks tika izmantots *Bluetooth* BLE bāka, pielietojot *Apple iBeacons* ietvaru, lai nodrošinātu iespēju palielināt bāku skaitu izvairoties no iespējamām problēmām identificēšanā.

Risinājuma izstrādes laikā autors saskārās ar apgrūtinājumiem, kuri izveides laikā tika atrisināti:

- Sākotnēji mobilā lietotne tika izstrādāta lietojot *React Native* ietvaru un programmēšanas valodu *Javascript*, taču progress bija lēns, kā arī tika sastapta problēma ar *Android Gradle*, kur bāku nolasīšanas izmantotajam ietvaram *react-native-beacons-manager* bija versiju nesakritība. Pēc saskaršanās ar šo problēmu, tika nolemts lietotni pārtaisīt, lietojot *Android Native* valodā *Kotlin* un bāku atkodēšanas ietvaru *AltBeacons*[19].
- Problēmas sagādāja projektā pieņemtā galvenā datubāze, kura tika veidota lietojot *MongoDB*. Šai datubāzei ir iebūvēts ietvars, kurš palīdz ar ārējiem pieprasījumiem, taču to nav iespējams savienot ar *Java* mobilo versiju, tādēļ radās nepieciešamība rast veidu kā datus no galvenās datubāzes nogādāt līdz galalietotājam un ievāktos datus par telpā esošajiem cilvēkiem nogādāt līdz projekta galvenajai datubāzei, lai tajā tiktu apkopota visa informācija. Lai nodrošinātu datu nogādi līdz lietotājam, Juris Veldre izveidoja API, kas pēc pieprasījuma atgriež datus par konkrēto telpu. Lai risinātu problēmu par telpā esošo cilvēku datu nogādi līdz galvenajai datubāzei, tika pielietotas *Firebase* mākoņfunkcijas un *Node.js* ietvars ‘*mqt*’. Ar šī ietvara palīdzību tika izveidots savienojums starp *Firebase* mākoņservisu un MQTT brokeri, pa kuru tika nosūtītas telpā esošo cilvēku datu izmaiņas.

Darba ietvaros tika uzstādīti uzdevumi, izpētītas mobilās lietotnes un cilvēku monitorēšanas iespējas. Tika izveidots risinājums, kas izmanto labās prakses piemērus un izpilda uzstādītos uzdevumus.

## Izmantotā literatūra

1. Waze datu ievākšanas dokumentācija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://www.waze.com/legal/privacy#information-that-is-being-collected>
2. TikTok datu ievākšanas informācija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://medium.com/@fs0c131y/tiktok-logs-logs-logs-e93e8162647a>
3. Bluetooth zema spēka bāku konfigurēšanas iespējas, specifikācija un definīcija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: [https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/#:~:text=Bluetooth%20Low%20Energy%20\(BLE\)%20Beacons,are%20closed%20and%20cost%20money.](https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/#:~:text=Bluetooth%20Low%20Energy%20(BLE)%20Beacons,are%20closed%20and%20cost%20money.)
4. Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project) projekta informācija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://www.stradini.lv/lv/content/jaunas-tehnologijas-covid-19-pacientu-temetai-monitoresanai-testesanai-un-terapijai-3-t>
5. *Firebase* dokumentācija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://firebase.google.com/>
6. Sistēmas github koda glabātuve. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://github.com/JurisVeldre/covid-monitoring-app>
7. Mosquitto mājaslapa. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://mosquitto.org/>
8. Apraksts par NoSQL lietošanu. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://www.couchbase.com/resources/why-nosql>
9. Vienas kameras cilvēku monitorēšanas princips. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://www.koreascience.or.kr/article/CFKO200211921321797.page>
10. Vairāku kameru cilvēku monitorēšanas principi. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <http://www.cs.cornell.edu/~rdz/Papers/KZ-ICMS99.pdf>
11. iBeacons bāku ietvara struktūra. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams: <https://support.kontakt.io/hc/en-gb/articles/201492492-iBeacon-advertising-packet-structure#:~:text=iBeacon%20packet%20structure&text=UUID%20is%2016%20bytes%20long,calculate%20distance%20from%20the%20iBeacon.>
12. Vairāku *Bluetooth* bāku ietvaru specifikācija. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://austinblackstoneengineering.com/ble-beacons-ibeacon-altbeacon-uribeacon-and-derivatives/>

13. Eddystone bāku ietvara struktūra. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://www.mokosmart.com/eddystone-protocol-and-specifications/>

14. Cilvēku skaita novērtējums izmantojot CO2 sensoru. [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

[https://www.researchgate.net/publication/261502167\\_Estimation\\_of\\_the\\_number\\_of\\_people\\_under\\_controlled\\_ventilation\\_using\\_a\\_CO2\\_concentration\\_sensor](https://www.researchgate.net/publication/261502167_Estimation_of_the_number_of_people_under_controlled_ventilation_using_a_CO2_concentration_sensor)

15. React Native dokumentācija [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://reactnative.dev/>

16. Kotlin un Java programmēšanas valodu salīdzinājums [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://www.xenonstack.com/blog/kotlin-android/#:~:text=Kotlin%20Application%20Deployment%20is%20faster,be%20executed%20in%20the%20JVM.>

17. MQTT priekš Node.js [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://www.npmjs.com/package/mqtt>

18. Altbeacons dokumentācija [tiešsaiste - pārbaudīts 30.05.2021] Pieejams:

<https://altbeacon.github.io/android-beacon-library/>

Bakalaura darbs „Sabiedrības virzīta mobilā lietotne Covid-19 infekcijas riska mazināšanai” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti.

Autors: Edgars Baumanis 31.05.2021. \_\_\_\_\_

Rekomendēju darbu aizstāvēšanai

Vadītājs: profesors, Dr. dat.zin. Leo Seļāvo 31.05.2021.

Recenzents: pasniedzējs, Mg. dat. Emīls Sjundjukovs

Darbs iesniegts Datorikas fakultātē 31.05.2021.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sprōģe

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

\_\_\_.06.2021. prot. Nr. \_\_\_\_.

Komisijas sekretārs: Dr. dat. **Ivo Odītis** \_\_\_\_\_