



VPP
Valsts pētījumu
programma



Projekts Nr. VPP-COVID-2020/1-0025

**Jaunās tehnoloģijas COVID-19 pacientu tēmētai monitorēšanai,
testēšanai un terapijai (3-T Project)**

Projekta papildu (specifiskie) rezultāti Nr.18

**Mobilā lietotne telpu apmeklējuma un
riska indikācijai**

Tehnoloģijas apraksts

8.pielikums

Autors: Edgars Baumanis

Rīga, 2020

Saturs

Saturs

Saturs	2
Ievads un problēmas apraksts	3
Risinājuma apraksts.....	4
Implementācija.....	6
Bluetooth low energy (BLE) bāka	7
Mobilā lietotne covid-monitoring-app	7
Firebase	10
Autorizācija	11
Tiešsaistes datubāze	11
Mākoņfunkcijas	11
Secinājumi	12
Izmantotā literatūra	13

Ievads un problēmas apraksts

Valdības projektā ar nosaukumu “Jaunās tehnoloģijas Covid-19 pacientu tēmētai monitorēšanai, testēšanai un terapijai (3-T Project)” [5] radās nepieciešamība rast risinājumu priekš viena no t - monitorēšanai (tracking).

Monitorēšana jāveic ar minimālu cilvēku traucējumu. Jāņem vērā arī, ka risinājums nevar būt pārāk plašs (piemēram WiFi), lai būtu iespēja atdalīt blakusesošas telpas.

Risinājuma apraksts

Risinājuma meklēšanas laikā tika apskatīti dažādi veidi, kā veikt cilvēku uzskaiti telpā. Viens no variantiem bija to veikt lietojot wifi tīklu, taču šī ideja netika lietota, jo wifi tīkli pārsež pārāk lielu distanci. Tādēļ par ideālo risinājumu priekš telpas izmēra laukuma noklājuma tika pieņemta Bluetooth low energy (BLE) bāka.

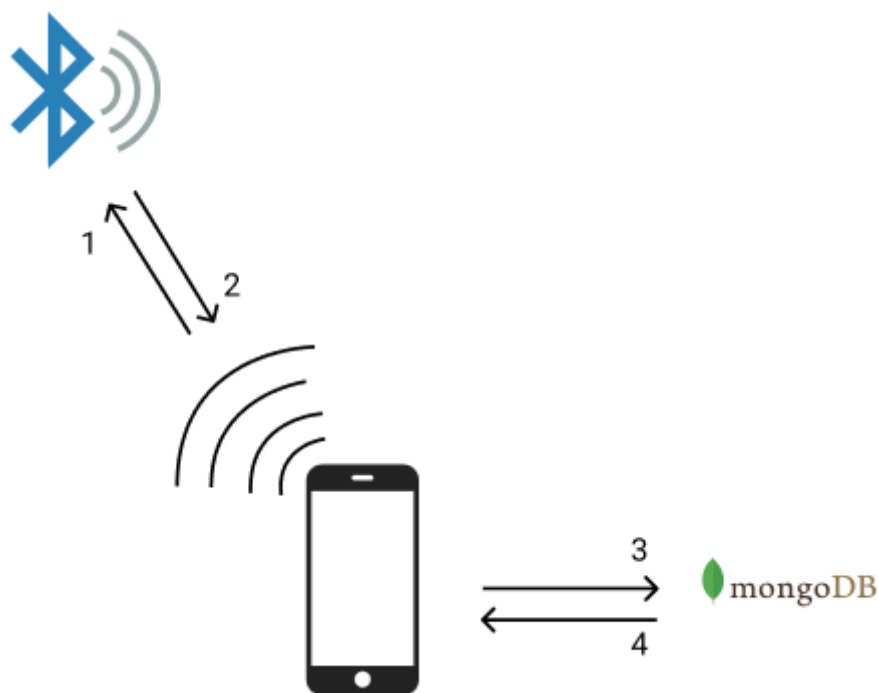
Bākas tiek izvietotas telpā, lai nosegtu maksimāli lielu reģionu un garantētu pārredzamību no visām telpas nostūrēm.



Attēls 1. Bluetooth zemas enerģijas bāku pārklājums

Risinājuma ietvaros bija arī nepieciešams atrast veidu kā cilvēkiem sazināties ar bākām. Šim nolūkam tika izplānota un izstrādāta mobila lietotne. Sākotnēji lietotne tika izstrādāta lietojot React Native, taču saskaroties ar problēmām, tika sāks darbs pie Android natīvas lietotnes.

Problēma, kas radās izstrādājot mobilo lietotni bija savienojums ar datubāzi MongoDB, jo no Android tiešu savienojumu uz MongoDB datubāzi, tāpēc tika izmantots Jura Veldres izstrādāts API, ar kuru iespējams veikt pieprasījumu uz noteiktu telpu un iegūt telpas mērījumus.



Attēls 2. Ieplānotā cilvēku skaitīšanas struktūra.

Attēlā 2 redzams ieplānotā cilvēku skaitīšanas vispārējā struktūra. Struktūrā redzams bluetooth bāka, mobilā lietotne, MongoDB datubāze.

1. Mobilā lietotne izraida signālu meklējot apkārtējās iBeacons tipa bluetooth bākas
2. iBeacon bluetooth bākas visu laiku raida savu klātesamību
3. Mobilā lietotne, ieraugot pierēģistrētu bāku sūta pieprasījumu uz MongoDB datubāzi, tajā ieliekot bākas ID, lai norādītu telpu, kurā cilvēks atrodas
4. MongoDB atgriež pieprasītos datus un pierēģistrē cilvēku bākas klātienē

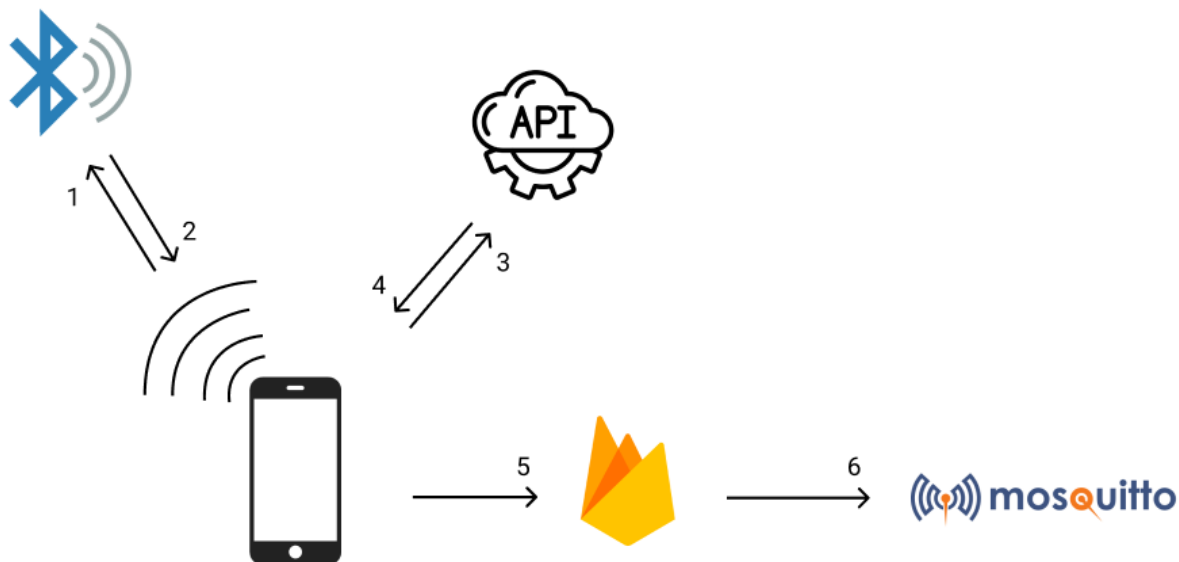
Šādai struktūrai parādījās divas ļoti apjomīgas problēmas:

1. Cilvēku skaitīšanai nepieciešams veikt kādu cilvēku reģistrēšanu
2. MongoDB neatbalsta API pieprasījumu veikšanu

Implementācija

Tālāk tiks aprasktīts kā tika pieiets augstāk redzamajām problēmām un dots padziļinātāks ieskats katras izveidotās struktūras galvenajā kodā un grūtības, kuras tika sastaptas.

Sākumā tiks aplūkots kā pieiets jau zināmām problēmām



Attēls 3. Izvēlētā cilvēku skaitīšanas struktūra

Attēlā 3 redzams cilvēku skaitīšanas vispārējā struktūra. Struktūrā redzams bluetooth bāka, mobilā lietotne, Jura Veldres izstrādātais API, Firebase, mosquitto mqtt brokeris.

1. Mobilā lietotne izraida signālu meklējot apkārtējās iBeacons tipa bluetooth bākas
2. iBeacon bluetooth bākas visu laiku raida savu klātesamību
3. Mobilā lietotne, ieraugot pierēģistrētu bāku sūta pieprasījumu uz API, tajā ieliekot bākas ID
4. API sūta pieprasītos datus
5. Lietotne nosūta paziņojumu Firebase par to, ka ir noteiktās bākas apkārtne
6. Firebase nosūta paziņojumu caur MQTT protokolu uz Mosquitto brokeri

Šeit parādās 3 jaunas lietas - API, Firebase un mosquitto brokeris.

- API tika izvēlēts kā risinājums, lai iegūtu datus no telpas sensoriem, vairāk par to var redzēt Jura Veldres aprakstā, kur viņš ieiet detaļās par to kā tas tiek nodrošināts un nodrošināts.

- Firebase tika izvēlēts kā risinājums priekš cilvēku uzglabāšanas. Firebase nodrošina drošu reģistrēšanās iespēju, kā arī lietojot Cloud Functions ir iespējams veikt savienojumu pār mqtt protokolu, kuru detalizētāk apraksta Valeriya Khlus savā aprakstā. Firebase tika arī lietota tiešsaistes datubāze, lai veiktu precīzu cilvēku uzskaitījumu un apstrādāt tālāk pieminēto individuālo ziņojumu par cilvēku skaitu
- Mosquitto MQTT brokeris ir sākums MongoDB datubāzes realizācijai, caur kuru iet visi dati un tiek iedalīti attiecīgi. Par šo vairāk var uzzināt Kalvja Sūniņa-Biedra aprakstā.

Tālāk tiks detalizēti aprakstīti visi izveidotie aspekti.

Bluetooth low energy (BLE) bāka

Tika eksperimentēts ar dažāda veida bāku simulātoriem, piemēram ESP32 konfigurētām, lai tas raida bluetooth signālu kā bluetooth bāka. Vēl priekš testēšanas tika izmantota mobila lietotne, kas ierīci pārtaisa par raidītāju.

Bākām nav pārāk daudz konfigurējamu iespēju. No iespējām, kuras attiecas uz datiem - UUID, Major, Minor, Tx Power šīs visas lietas ir iespējams konfigurēt katram piešķirot noteiktu datu tipu, noteiktā diapazonā.

- UUID - 32 heksadecimālu simbolu virkne (36 ar domuzīmēm)

Piemērs: fc956488-0860-45bc-9477-e11438ebd7b4

- Major - vesels skaitlis diapazonā no 0 līdz 65535
- Minor - vesels skaitlis diapazonā no 0 līdz 65535
- Tx Power - vesels skaitlis diapazonā -128 līdz 127

Tx Power izmanto, lai noteiktu cik tālu bāka pārsūta (-128 būtu tuvu raidoša bāka, 127 būtu tālu raidoša bāka)

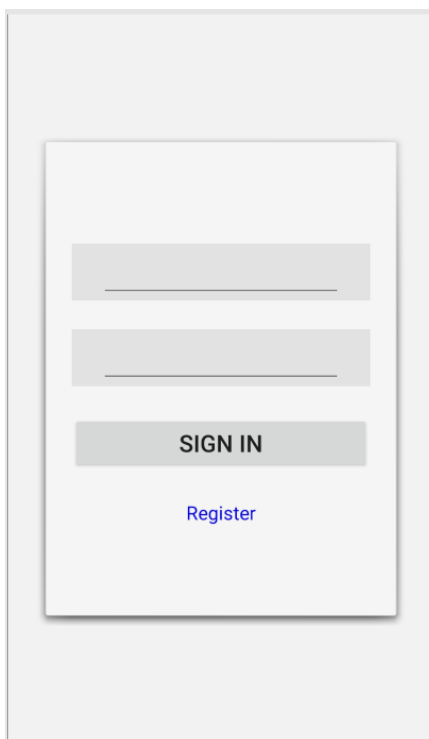
Mobilā lietotne covid-monitoring-app

Mobilās lietotnes gala produkts tika izstrādāts Kotlin programmēšanas valodā. Lietotnei tika izstrādāta lietotājsaskarne grafiskās izstrādes vidē Figma, pēc kuras tika taisīta gala lietotne.

Lietotnei tika izveidotas divas versijas, viena izmantojot React Native, taču to izstrādājot tika atrasta problēma ar izmantoto bluetooth bāku skenera un Firebase API instalācijām. Šis, kā

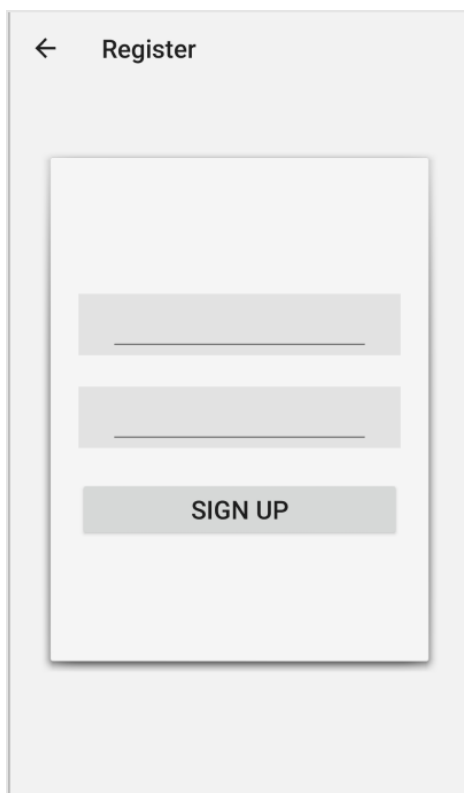
arī izstrādāšanas ātrums bija iemesli, kāpēc izstrādes vidū lietotne tika pārtaisīta natīvi uz Android sistēmas.

Lietotnes pamatā ir Android Beacon Library, kura adaptēta no primārā pielietojuma - AltBeacon bāku formātu meklēšanas uz iBeacon bāku formāta meklēšanu. Lietotnei ir trīs skati - Autorizācijas, Registrācijas, Galvenais datu pārskatīšanas.



Attēls 4. Autorizācijas skats

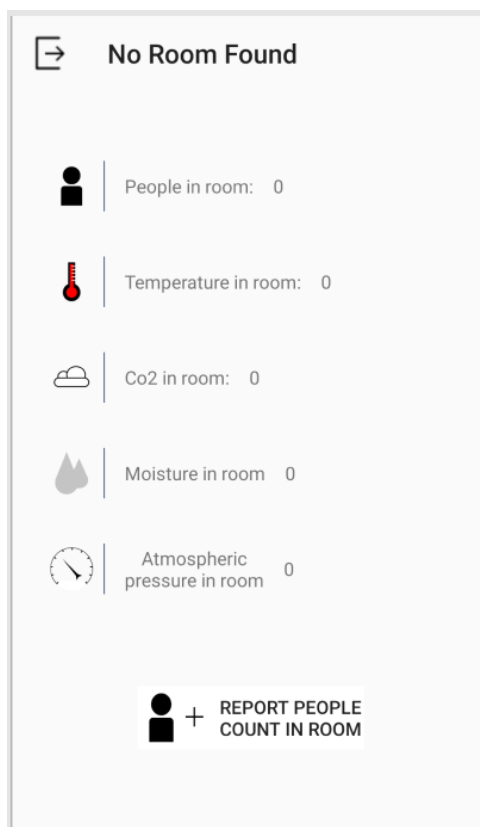
Attēlā 4 redzami divi lauki - viens priekš epasta, otrs priekš paroles. Poga paņem ievadītos datus un aizsūta tos uz Firebase Autorizāciju, kas nodrošina drošību. Register poga lietotāju aizved uz Registrācijas skatu (skat. 5. attēlu)



The image shows a mobile application interface for registration. At the top left, there is a back arrow and the text 'Register'. Below this, there is a central white box containing two input fields for text entry, one above the other. Below the input fields is a grey button with the text 'SIGN UP' in white capital letters.

Attēls 5. Reģistrācijas skats.

Reģistrācijas skatā ir divi lauki - augšējais priekš epasta, apakšējais priekš paroles. Nospiežot pogu “sign up” lietotājs tiek pierēģistrēts Firebase Auth sistēmā.



Attēls 6. Galvenais datu pārskatīšanas skats

Attēlā 6 redzams ko lietotājs redz, kad ir autorizējies sistēmā. Šajā ekrāna lietotājam ir iespēja redzēt attiecīgās telpas cilvēku skaitu, temperatūru, co2, mitrumu un atmosfērisko spiedienu. Skata pašā augšā redzams telpas nosaukums.

Telpas apakšā iespējams norādīt atlikušo cilvēku skaitu telpā. Kad lietotājs pietuvojas bākai, viņš tiek pierēģistrēts telpā, ar ID, kurš sakrīt ar bākas UUID. Kad lietotājs iziet no bākas rādiusa, tas tiek atreģistrēts no istabas, kā arī viņa cilvēku skaita ziņojumi no istabas tiek izdzēsti.

Firestore

Firestore ir mākoņserviss, kas atvieglo mobilo un tīmekļbāzētu lietotņu izstrādi nodrošinot tās ar viegli uzstādāmu datubāzi un API.

Firestore šajā projektā spēlē trīs daļas:

- Autorizācija
- Mākoņfunkcijas
- Tiešsaistes datubāze

Autorizācija

Firestore autorizācija ir google izveidots rīks, kas sagādā drošu un ātru veidu kā reģistrēt, pieglabāt un autorizēt lietotājus. Šim projektam tiek iespējota viena no daudzām autorizācijas iespējām - epasta un paroles autorizācija. Šī iespēja ļauj lietotājam ievadot epastu un paroli pieslēgties sistēmai.

Tiešsaistes datubāze

Firestore tiešsaistes datubāze (real time database) ir no-sql datubāze, kas nozīmē, ka tajā nav relāciju, bet visa struktūra tiek veidota uz atslēgas - vērtības (key - value) principa. Datubāzē ir iespējams veidot mezglus, kas var saturēt citus mezglus un atslēgas - vērtību pārus.

Šī datubāze tiek izmantota, lai atdalītu lietotāju moduļa funkcijas no galvenās sistēmas (MongoDB) un to lieki nenoslogotu. Datubāze ir atbildīga par telpu un cilvēku saiti. Tā uzglabā kurā telpā (pēc ID) kurš reģistrēts lietotājs atrodas un to, cik kurā telpā atrodas cilvēki balstoties uz reģistrēto cilvēku ziņojumiem.

Mākoņfunkcijas

Firestore mākoņfunkcijas (cloud functions) strādā tandēmā ar tiešsaistes datubāzi, reaģējot uz izmaiņām tajā. Kopā tiek lietotas 3 cloud funkcijas -

- viena, kas reaģē uz lietotāju istabas maiņu,
- viena, kas reaģē uz istabas ziņojumu skaita maiņu
- viena, kas reaģē uz istabas cilvēku skaita maiņu

Pielietojot šīs trīs funkcijas iespējams maksimāli automatizēt lietotāja pieredzi.

Secinājumi

Uzdevuma ietvaros tika izstrādāts mobilais risinājums kas ietver gan mobilo lietotni gan arī Firebase datu bāzi lietotnes atbalstam un autentifikācijai. Lietotne sadarbojas ar sensoru datu bāzes API, kas nodrošina informāciju par vides, piemēram, telpas stāvokli, tai skaitā temperatūru, mitrumu, CO2 līmeni gaisā un infekcijas risku. Lietotne nodrošina arī cilvēku skaita noteikšanu telpā vai zonā ar divām metodēm: pirmkārt, ar BT bāku palīdzību, kas izvietotas mērķa telpā, lietotne automātiski reģistrē lietotāja klātbūtni telpā, un otrkārt, lietotājam ir iespēja iesūtīt aptuvenu cilvēku skaitu telpā, kas var tikt izmantots lai noteiktu cilvēku skaitu un līdz ar to infekcijas risku telpā. Lietotni iespējams papildināt arī ar citu funkcionalitāti, piemēram, notifikācijām par augsta CO2 līmeni telpā un ieteikumu vēdināšanai. Izstrādātās programmatūras pieejams GitHub repozitorijā <https://github.com/JurisVeldre/covid-monitoring-app>

Pēc testiem, kas tika veikti ar simulētām BT bākām, kā arī pārskatot vairākus rakstus, kur tika pētīta tieši BLE pārraides jauda pret attālumu, kuru tās pārraida, var secināt:

- BLE bākas, neatkarībā no standarta, viegli ietekmējas no WiFi, kas samazina to pārraidspēju par apmēram 75% [1]
- Bāku pārraidi ietēmē objekti telpā (piemēram, ja tā ir cilvēku līmenī, tad tai būs daudz traucējumu, tāpēc to labāk novietot augstāk u. tml.)

Jāpiebilst, ka dažādu izstrādātāju bākas atšķirās pārraides spēkā. Simulētās bākas, kuras tika izmantotas priekš testēšanas, balstījās uz telefona Bluetooth pārraides spēku.

Darba problēma tika atrisināta, izstrādājot sistēmu, kas veic cilvēku uzskaiti. Tomēr, tika atrasti ietekmējošie faktori, kas var ietekmēt sistēmas efektivitāti.

Izmantotā literatūra

1. A Measurement Study of BLE iBeacon and Geometric Adjustment Scheme for Indoor Location-Based Mobile Applications [tiešsaiste – pārbaudīts 18.12.2020] Pieejams: <https://www.hindawi.com/journals/misy/2016/8367638/>
2. Transmission power, Range and RSSI [tiešsaiste – pārbaudīts 18.12.2020] Pieejams: <https://support.kontakt.io/hc/en-gb/articles/201621521-Transmission-power-Range-and-RSSI>
3. Firebase dokumentācija [tiešsaiste - pārbaudīts 18.12.2020] Pieejams: <https://firebase.google.com/docs>
4. Kotlin dokumentācija [tiešsaiste - pārbaudīts 18.12.2020] Pieejams: <https://kotlinlang.org/docs/reference/>
5. 3-T Project [tiešsaiste - pārbaudīts 18.12.2020] Pieejams: <https://lzp.gov.lv/project/3-t-project/>