



RELAZIONE PROGETTO BUBBLEBOX

Relazione e manuale di utilizzo device e app
BubbleBox

SUNTO

Sistema di rilevamento di prossimità per prevenire e rilevare l'espansione del COVID-19 (e altre forme virali), attraverso l'uso di un device e di un App (non essenziale, ma che fornisce funzionalità aggiuntive) smartphone e web. Oltre all'analisi dei dati, questo progetto, effettua soprattutto una raccolta puntuale, completa, precisa e organizzata, con dati pronti per essere analizzati, infatti, il device effettuerà un vero proprio Edge Computing, in modo da gestire lui stesso i dati e presentarli, attraverso più canali comunicativi (Bluetooth smartphone o WiFi), al sistema server. Il nome BubbleBox proviene proprio dall'idea di una bolla virtuale attorno ad ogni individuo che, se violata, da quella di un altro, ottiene ID di quest'ultimo (in modo sicuro e anonimo) e invia la segnalazione ad un server online, dopo essere stato elaborato, per analisi e verifiche future. Se un individuo dovesse essere positivo al COVID-19 allora verranno allertate e controllate tutte le persone con cui è entrato in contatto senza ambiguità alcuna. A differenza di alcuni sistemi di tracciamento già presenti, il nostro fornisce un servizio funzionante in ogni luogo e offline, perché la raccolta avviene prima in locale poi, quando disponibile, viene inoltrata online.

Pietro Rignanese e Andrea Polenta
Sistemi Operativi Distribuiti Tempo Reale

Sommario

INTRODUZIONE.....	2
UTILIZZI	4
ARCHITETTURA E FUNZIONAMENTO	6
1. <i>ARCHITETTURA DEVICE</i>	6
2. <i>FUNZIONAMENTO DEVICE</i>	9
3. <i>CIRCUITO DEVICE</i>	12
4. <i>CONSUMI DEVICE</i>	16
5. <i>APP SMARTPHONE E WEB APP</i>	17
5.1. <i>Gestione dei dati</i>	19
6. <i>FUNZIONAMENTO DEVICE + APP</i>	22
7. <i>FUNZIONAMENTO APP</i>	23
TECNOLOGIE DEVICE	24
RACCOLTA DATI (Questionario).....	25
CONCLUSIONI	28
 MANUALE D'USO.....	 31
1. <i>INIZIALIZZAZIONE DEVICE</i>	31
2. <i>FIRMWARE SCHEDE</i>	33
3. <i>APP</i>	35
4. <i>MOCKUP DISPLAY DEVICE</i>	38

INTRODUZIONE

Cos'è **BubbleBox** e perché si è voluto dare questo nome a questo progetto universitario. Le domande che ci si pone sono tante, anche come affronteremo l'emergenza "Coronavirus" o altre future epidemie. Questa proposta, appoggiata dal professor Aldo Franco Dragoni, dell'università politecnica delle marche e dal suo laboratorio **AIRTLab**, nasce dall'idea di poter gestire, controllare la diffusione di una qualsiasi forma di virus e patologia, raccogliendo i dati, tra tutti gli utenti, in modo *puntuale ed efficiente*. L'intera elaborazione e raccolta dei dati avviene in *locale direttamente sul device* (che ha l'aspetto di un normalissimo orologio da polso), in modo da *non sovraccaricare troppo il sistema server*. I dati da mandare al server viaggiano attraverso canali comunicativi semplici da utilizzare perché utilizzati costantemente ogni giorno (da chiunque o dalla stragrande maggioranza delle persone), tra cui la connessione allo smartphone, tramite Bluetooth, e la connessione ad una rete WiFi attraverso il bottone WPS, qualora fosse disponibile, o la normalissima connessione con SSID e Password, inoltre i dati viaggiano su protocolli di comunicazione leggeri di tipo **publish-subscribe** chiamato **MQTT** (*Message Queue Telemetry Transport*) che ha un bassissimo impatto e richiede pochissima banda.

Oggi, Aprile 2020, siamo in piena emergenza COVID-19, ci sono ancora tantissimi casi di contagi, anche se il numero dei malati guariti supera quello dei morti, le vite che sono state portate via superano i 20 mila e questo numero continua a salire, la paura rimane e le persone sono costrette a rimanere in casa e adottare misure molto restrittive e precauzionali. ***Cosa succederà quando dovremo cominciare a tornare alle nostre normali vite e frequentare di nuovo luoghi affollati?*** Il nostro progetto è mirato a garantire un'ulteriore *misura di sicurezza*, oltre a quelle da rispettare, ma che riesce a gestire l'intera emergenza in modo **poco invasivo e restrittivo**.

Questa idea di progetto nasce nel Febbraio 2020, quando, a due studenti (Pietro Rignanese e Andrea Polenta), del corso magistrale di "Ingegneria informatica e dell'automazione" dell'UNIVPM, viene in mente che si potrebbe realizzare un device apposito per poter monitorare il virus o qualsiasi forma virale (il device è molto versatile e potrebbe essere utilizzato in moltissimi ambiti, da quello sanitario, aziendale, per monitorare gli accessi in aree o limitare il numero di accessi ecc...). Il nome BubbleBox sta proprio a immaginare una bolla protettiva attorno ad ogni persona, una bolla che "SCOPPIA" quando la sua sfera personale e privata viene a contatto con quella di un'altra persona.

Attraverso degli studi, e per le normative di legge in vigore dal Marzo 2020, le persone *devono distanziarsi di almeno un metro uno dall'altro*. Questa normativa non sempre viene seguita alla lettera e risulta difficile in situazioni particolari, infatti, *ricordare tutte le persone che si sono incontrate in tutti i posti che si sono visitati negli ultimi 15 giorni risulta essere complicato e non tutti ricordano ogni singola persona incontrata*, anche perché si possono frequentare posti in cui ci sono persone che non si conoscono (bar, farmacie, pizzerie, supermercati, ecc...). Il BubbleBox serve proprio a questo: ***catturare ogni singolo contatto avvenuto tra due o più individui***, in modo che, quando uno di questi individui risulta essere infetto, si potrà risalire, in modo specifico e puntuale, a quelli che sono stati i suoi contatti negli ultimi 15-20 giorni e agire prontamente per isolare chi serve ed evitare la diffusione non controllata del virus. Le persone avranno i propri dati completamente oscurati agli altri e protetti, infatti si potrà conoscere solamente il numero di persone incontrate giornalmente, o negli ultimi 15 giorni, e non di chi si tratta proprio per *non infrangere la privacy di ogni singolo individuo*. ***La privacy è importante è il nostro progetto prevede di tutelarla attraverso ogni forma di gestione e controllo possibile***. I dati saranno disponibili, qualora servissero, solamente alla sanità, o chi gestirà l'emergenza, per poter eseguire le varie prevenzioni, controlli e tamponi.

Oltre all'analisi dei dati, questo progetto, effettua soprattutto una ***raccolta puntuale, completa e organizzata***, con dati pronti per essere analizzati (avere già dei dati pronti e privi di rumore, garantisce un'analisi più accurata, semplice e precisa). Se un individuo dovesse essere positivo al COVID-19 (o altri virus, in futuro) allora ***verranno allertate e controllate tutte le persone con cui è entrato in contatto senza ambiguità alcuna***. A differenza di alcuni sistemi di tracciamento già presenti, il nostro fornisce un servizio funzionante in ogni luogo e offline, perché *la raccolta avviene prima in locale poi, quando disponibile, viene inoltrata online attraverso varie metodologie*.

Il progetto non è limitato solo all'utilizzo del device, ma ci sarà a disposizione anche una App apposita (non essenziale ma che fornisce delle funzionalità in più) per smartphone o web per poter accedere ai propri dati e visualizzare le funzionalità da eseguire, come il numero di persone con cui si è entrati in contatto e la segnalazione delle patologie. Questi dati saranno utilizzati quando si verifica un contagio, in modo che sarà possibile risalire, andando a ritroso, alla persona malata che ha cominciato a diffondere il virus.

Si vive in tempi difficili ed è per questo che dobbiamo cambiare i nostri modi di comportarci e di analizzare i vari rischi. Questo device non fermerà l'avanzare del virus in questione, ma potrebbe dare una grossa mano a tante persone, specialmente medici e sanità, in modo da poter monitorare e gestire le varie emergenze e tenere sott'occhio l'avanzamento del virus e lo stato di salute delle persone a rischio.

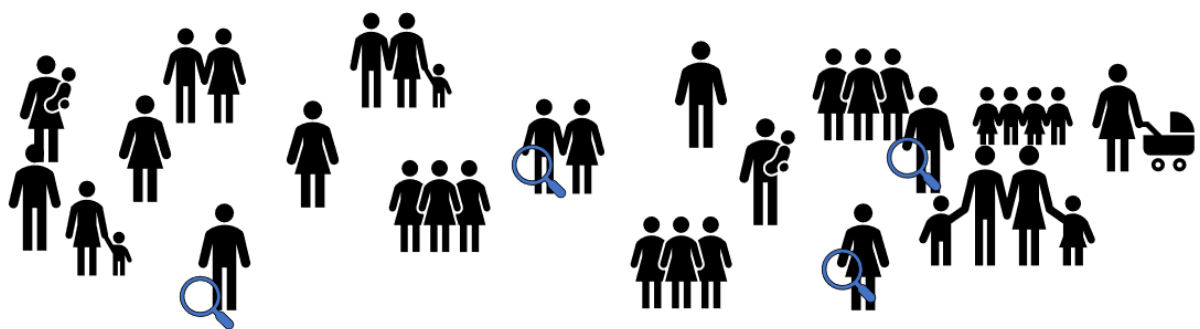
UTILIZZI

Gli utilizzi del device sono molteplici, oltre a raccogliere i dati in modo puntuale e preciso, questi dati possono essere utilizzati per:

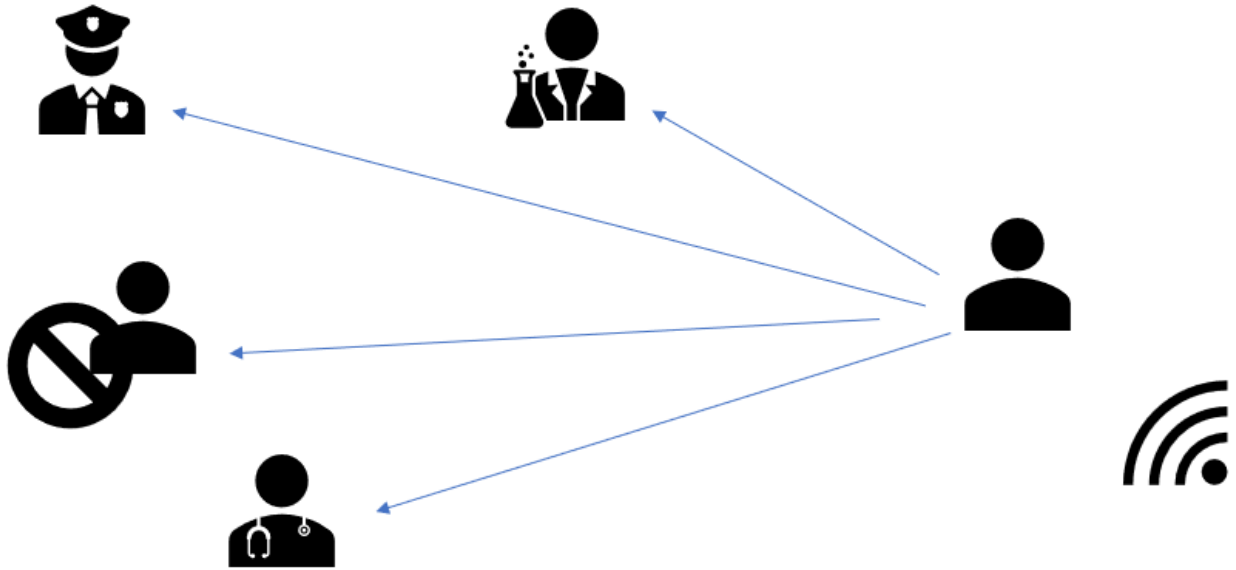
- Sapere subito *quali sono le persone entrate in contatto con un infetto* o un probabile infetto in modo da intervenire prontamente
- Analisi dei dati per verificare la *diffusione e il tracciamento del virus*



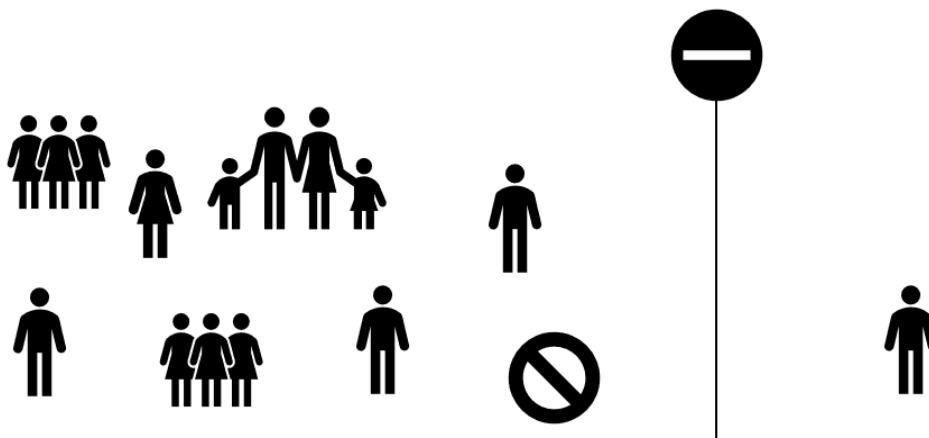
- Sapere su quali persone agire per effettuare dei “Tamponi campione” in modo da controllare un numero piccolo di persone ma in modo efficiente



- *Controllare in modo veloce l'identità delle persone e il perché sono in giro anche se vice la regola "RESTA A CASA"*



- *Controllare gli accessi in determinati luoghi e far accedere solo persone addette, o far accedere un certo numero di persone*



ARCHITETTURA E FUNZIONAMENTO

1. ARCHITETTURA DEVICE

Il device è composto da tante piccole parti, ognuna delle quali svolge una particolare funzione:

- **ESP32 WROOM 32:**

- È un modulo che dispone sia di una componente WIFI (2.4 Ghz) e di una Bluetooth di tipo BLE (Bluetooth Low Energy)
- È molto piccolo e i suoi consumi si aggirano intorno ai 50 mAh quando è in funzione e 0.05 mAh quando è in fase di “sonno profondo” (deep sleep)



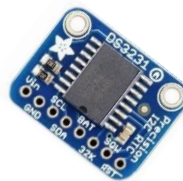
- **Display OLED 0.96”:**

- Display molto piccolo che serve a far visualizzare data, ora ed eventuali funzioni del device che è possibile utilizzare



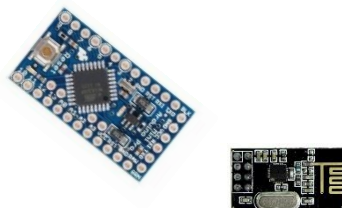
- **RTC DS3231:**

- Modulo che tiene il conto della data e ora



- **Arduino Micro + NRF24L01+:**

- Questi due moduli vengono utilizzati per scandagliare l’area alla ricerca di un altri device BubbleBox e svegliare, quando ne viene trovato uno nei dintorni, l’ESP32, questo perché il suo consumo è sotto i 15 mAh



- **Lettore MicroSD:**
 - Modulo per leggere una micro SD da utilizzare come memoria di massa



- **Pila litio**
- **Caricabatterie**



Questa struttura del device sarà la struttura finale. Ogni singola parte è grande come un'unghia o poco più, in modo che sarà possibile costruire un device leggero, facilmente trasportabile e che avrà le sembianze di un orologio da polso.

La struttura del device, per le varie prove e test, sarà leggermente diversa per motivi di comodità di sviluppo e praticità.

Ecco la struttura che andremo ad utilizzare per il prototipo ed effettuare i vari test:

- **DevKit ESP32:**
 - Modulo con chip ESP32 WROOM 32 impiantato
 - Facile da utilizzare e collegare con una breadboard e cavetti



- **Display OLED 0.96"**
- **RTC DS3231**

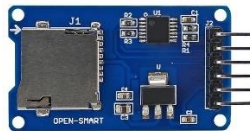


- **RF-NANO:**

- Modulo che racchiude Arduino Nano + NRF24L01
- Comodo da utilizzare e leggermente più grande del Micro
- Utilizzato per motivi di praticità

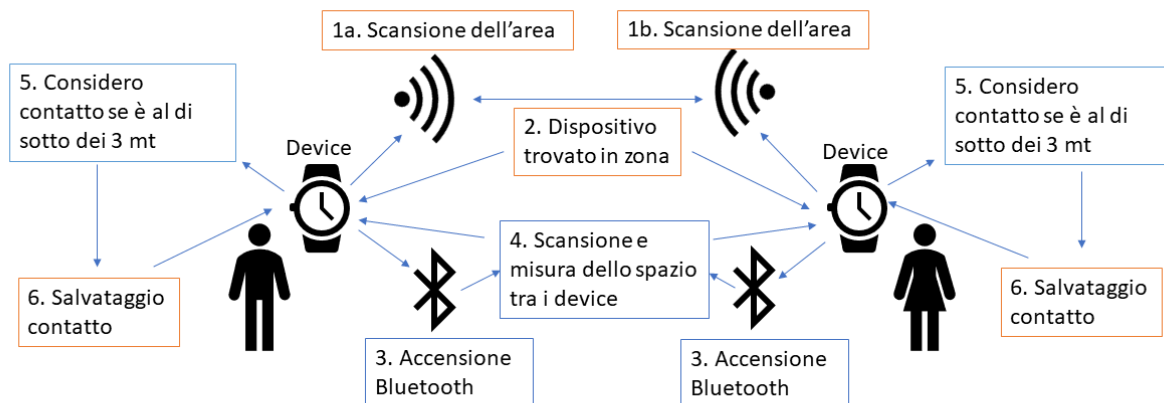


- **Lettore SD Card**



- **Alimentazione USB**

2. FUNZIONAMENTO DEVICE

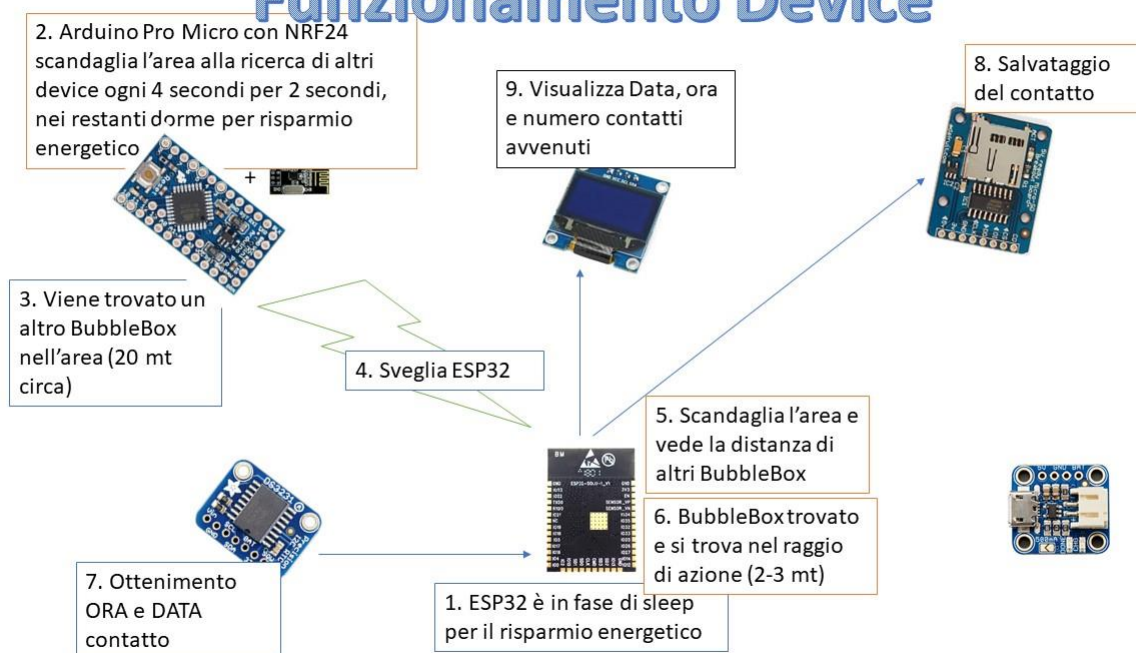


Come si può vedere dall'immagine riportata, il funzionamento del device avviene attraverso 6 fasi principali:

1. **Scansione dell'area:**
 - a. Viene scansionata l'area alla ricerca di altri device per poter comunicare
 - b. La scansione avviene attraverso il modulo Micro + NRF24L01 (o nel caso semplice RFNANO) perché consuma pochissima energia rispetto all'ESP32 e copre un raggio di 20 metri.
2. **Ritrovamento dispositivo:**
 - a. La scansione ha trovato in zona uno o più dispositivi BubbleBox di altre persone, questo significa che il dispositivo ESP32 deve entrare in funzione
3. **Accensione Bluetooth:**
 - a. Viene attivato il dispositivo bluetooth dell'ESP32
 - b. In questo momento il device mette a disposizione di tutti i dispositivi bluetooth le proprie informazioni (MAC, NOME e UUID)
4. **Scansione e misura spazio tra device BubbleBox:**
 - a. In questo momento l'ESP32 scansiona l'area e calcola la potenza dei segnali trovati per calcolare la distanza che li separa (RSSI)
5. **Contatto avvenuto:**
 - a. Se i device sono al di sotto della soglia da considerare (2 metri) scatta il contatto!
 - b. Considero il contatto
6. **Salvataggio contatto:**
 - a. Salvo, in locale, il contatto o i contatti avvenuti
7. **Spegnimento device:**
 - a. Spengo il device se non è presente alcun altro device BubbleBox nelle vicinanze per risparmiare energia.

Tutto il funzionamento può essere rappresentato nel dettaglio e su come le componenti interagiscono tra loro.

Funzionamento Device

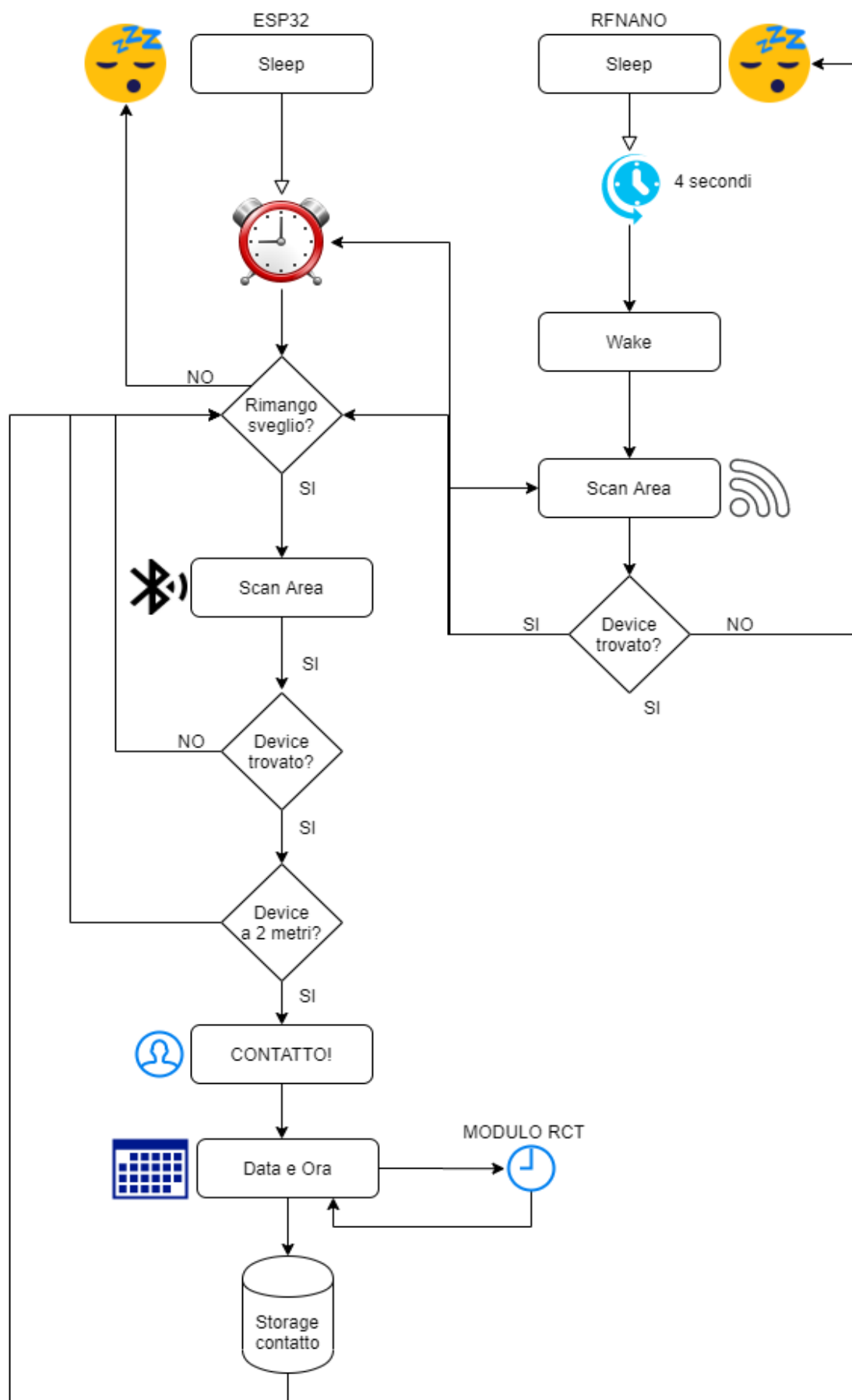


Come si può vedere dall'immagine, le fasi, per il funzionamento del device, sono molto più articolate e specifiche, in quanto, le componenti interagiscono fra loro per cambiarsi dati, salvarli e gestirli.

Le fasi sono le seguenti:

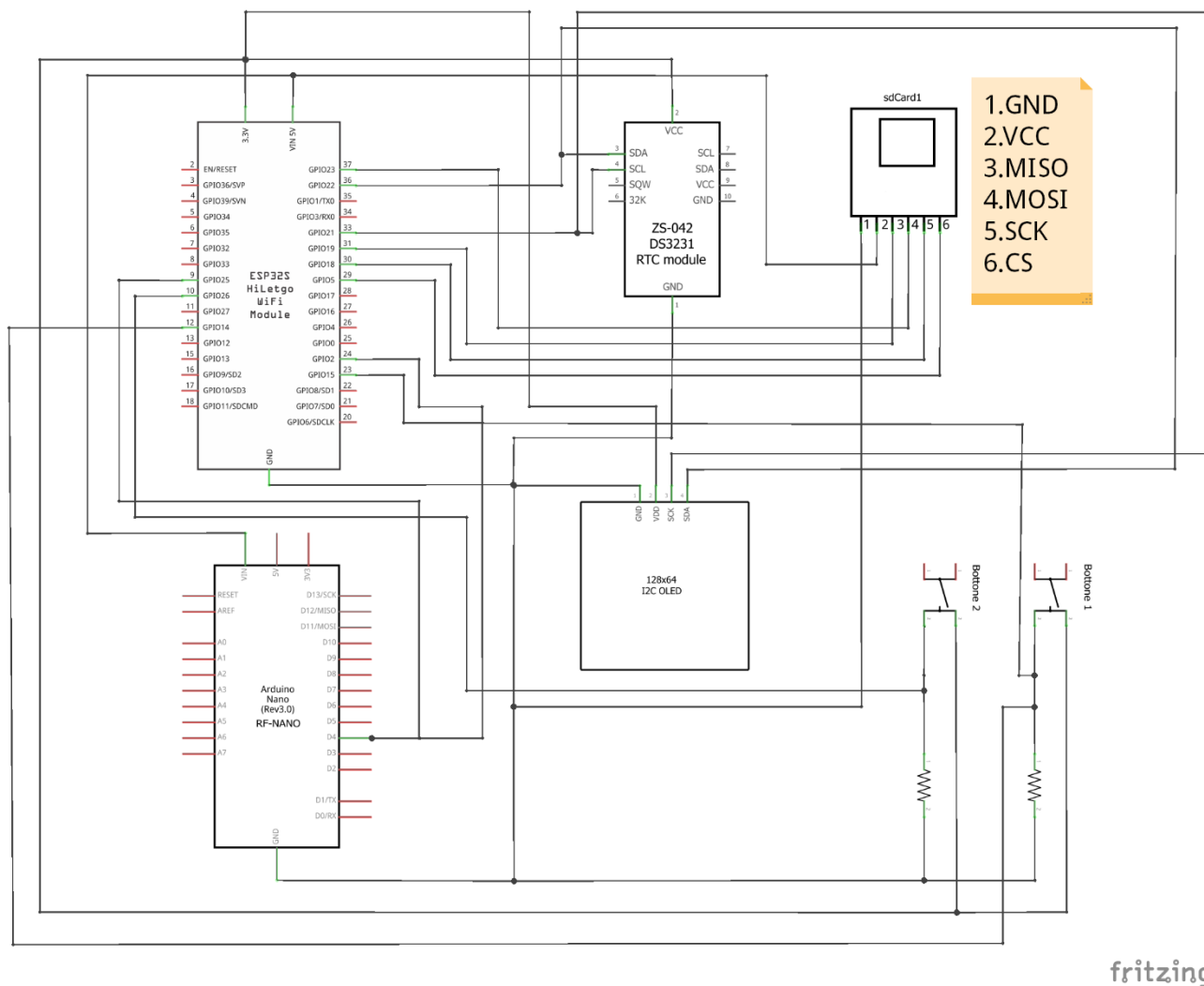
1. L'ESP32 è in fase di "sonno profondo" per risparmiare energia e non si sveglierà fino a che non sarà presente un altro device BubbleBox nelle vicinanze
2. Arduino Micro con NRF24L01 scandaglia l'area (in un raggio di 15-20 mt) alla ricerca di un altro device BubbleBox
 - a. La scansione avviene ogni 4 secondi e segue una fase di sonno profondo in quel lasso di tempo per poter risparmiare ancora energia
3. Viene ritrovato un altro device nelle vicinanze
4. Viene mandato un segnale di "WAKE" all'ESP32 per svegliarlo e tornare alle sue normali funzioni
5. L'ESP32 si sveglia, accende il suo Bluetooth e comincia a scandagliare l'area alla ricerca di un device BubbleBox
6. Se un device (BubbleBox) di un'altra persona viene trovato e si trova al disotto dei 2 metri, lo considera (si considera l'ID del device e non il nome o altri dati della persona, il completo anonimato è mantenuto per tutto l'iter)
7. Ottiene la data e l'ora dal modulo RTC
8. Salva il contatto sulla propria memoria
 - a. Il dato salvato avrà la seguente forma: DATA|ORA|MioID|IDContatto
 - b. Con MioID e IDContatto viene considerato il MAC address del bluetooth BLE ESP32

Tutte le fasi ed il funzionamento nello specifico è rappresentato dal seguente flowchart:



3. CIRCUITO DEVICE

Per definire il circuito, sono stati realizzati diversi schemi, in modo da rendere chiaro e preciso i vari collegamenti tra i moduli. Partiamo dal circuito specifico generale rappresentato dalla figura seguente:



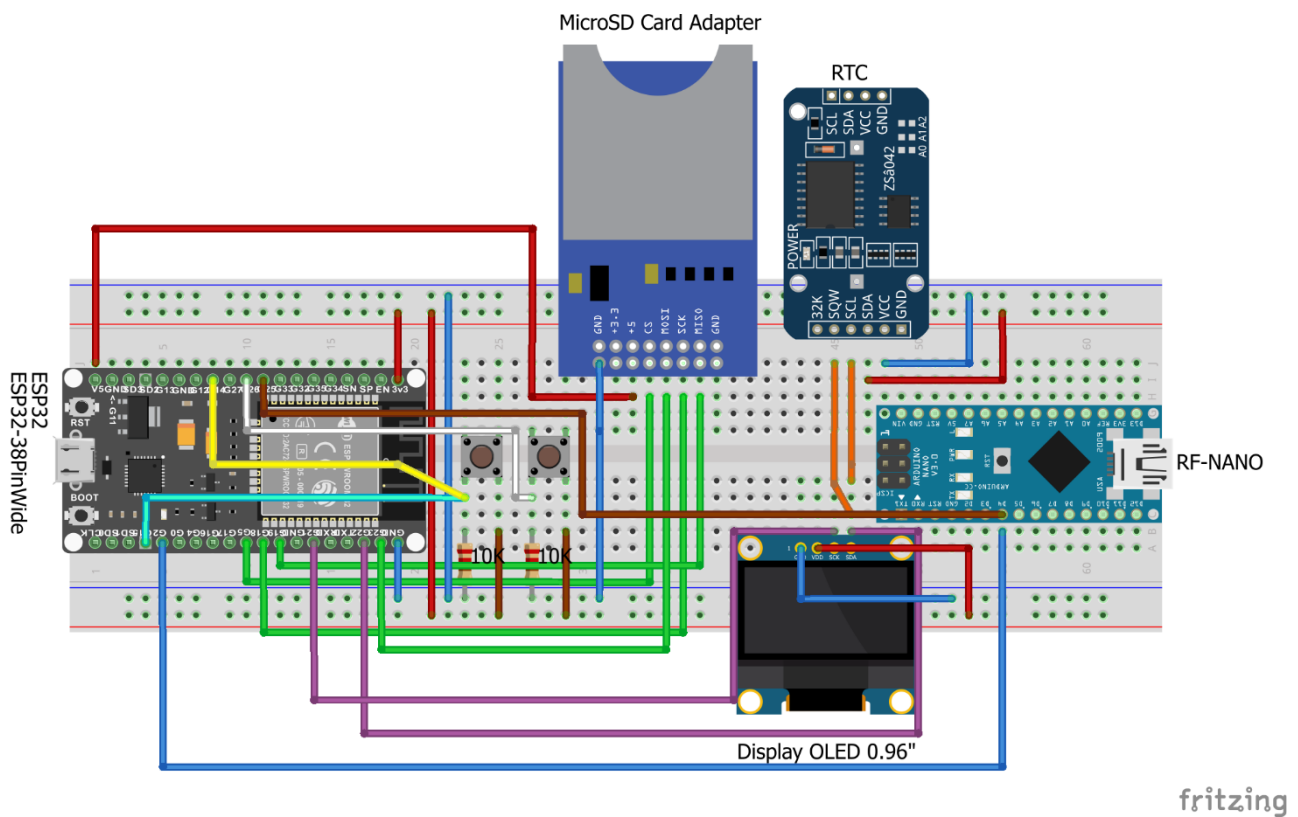
fritzing

Per tutte le prove, test e circuiti che verranno mostrati, saranno utilizzati i moduli di sviluppo (DEVKit) per semplicità di rappresentazione, costruzione circuito e funzionalità. Il tutto sarà poi replicato, una volta eseguiti i vari test, con i moduli più piccoli.

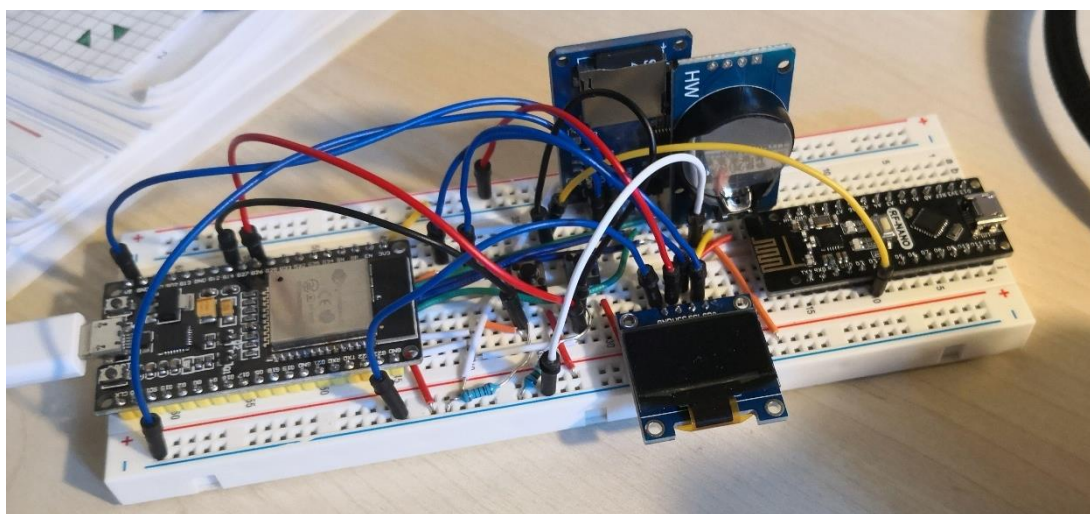
Il circuito che si sta osservando è molto specifico e si vedono i vari componenti nello specifico. Ogni singolo componente è collegato, principalmente, alla rete di alimentazione (3 Volt) e all'ESP32 che gestisce e orchestra l'intero sistema. Il display OLED sarà collegato ai due pin SDA e SCL dell'ESP32 che corrispondono, rispettivamente, a 22 e 21 (GPIO 22, GPIO 21), il modulo RTC, con i pin SDA e SCL, anch'esso ai pin 22 e 21 (questo non crea alcun problema all'ESP32, riuscendo a gestire due moduli anche se sono connessi agli stessi pin). Il modulo lettore di SD card, in questo caso, è alimentato a 5V, ma nel circuito finale sarà usato uno adatto con alimentazione a 3V, in modo da non creare problemi di alimentazione essendo che la pila al litio erogherà 3.7V. L'RFNANO sarà collegato all'ESP32 su due pin, uno per svegliarlo quando viene avvertito

un altro device nelle vicinanze e l'altro per tenerlo sveglio. Sono presenti due bottoni per poter accendere il display e selezionare le opzioni che compaiono.

Di seguito il circuito installato sulla millefiori:



Il seguente circuito è più semplice ed intuitivo del precedente e rappresenta una foto più dettagliata del prototipo con i collegamenti ben definiti. Infatti, come possiamo vedere, tale circuito è perfettamente paragonabile al circuito reale:

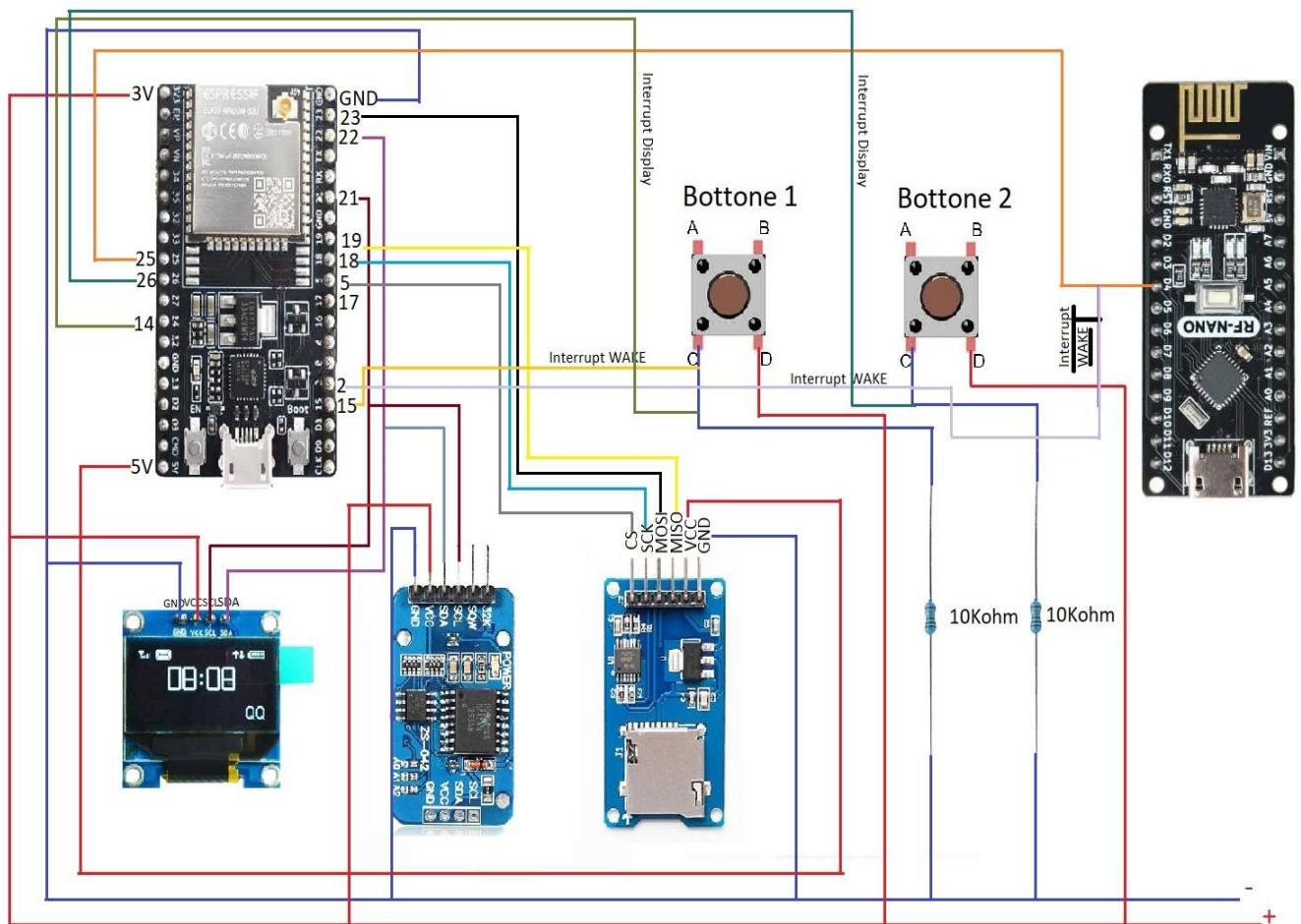


Nella foto, risulta molto più complesso capire i collegamenti, ma nello schema presedente i dettagli sono più definiti e sarà possibile ricreare il circuito sulla propria millefiori (breadboard).

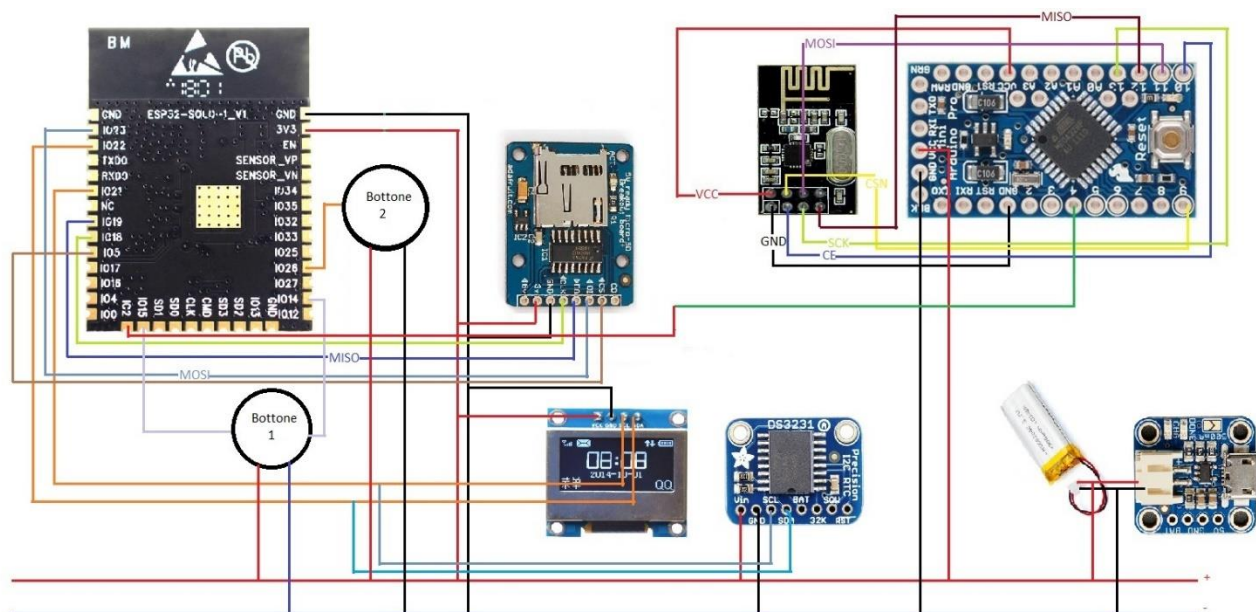
Per avere una linea guida sui collegamenti, di seguito è presente una tabella in cui è possibile vederli nel dettaglio:

ESP32	RFNANO	RTC	DISPLAY	SD CARD	Bottone 1	Bottone 2
<i>GPIO 25</i>	<i>D4</i>					
<i>GPIO 21</i>		<i>SCL</i>	<i>SCL</i>			
<i>GPIO 22</i>		<i>SDA</i>	<i>SDA</i>			
<i>GPIO 19</i>				<i>MISO</i>		
<i>GPIO 23</i>				<i>MOSI</i>		
<i>GPIO 18</i>				<i>SCK</i>		
<i>GPIO 5</i>				<i>CS</i>		
<i>GPIO 14</i>					<i>Cambio Display</i>	
<i>GPIO 26</i>						<i>Sel. Opz. Display</i>
<i>GPIO 2</i>	<i>D4 (WAKE)</i>					
<i>GPIO 15</i>					<i>WAKE</i>	
<i>3V</i>		<i>VCC</i>	<i>VCC</i>		<i>VCC</i>	<i>VCC</i>
<i>5V</i>				<i>VCC</i>		
<i>GND</i>		<i>GND</i>	<i>GND</i>	<i>GND</i>	<i>GND</i>	<i>GND</i>

E di seguito un ulteriore schema di connessione con ulteriori dettagli per i collegamenti:



Il tutto dovrà poi essere riconducibile a questo schema:



Grazie alla grandezza di ogni singolo modulo, sarà possibile implementare l'intero sistema all'interno di un orologio da polso (o qualcosa di molto piccolo e facilmente trasportabile).

4. CONSUMI DEVICE

Di seguito sono riportati i consumi, in termini di milliampere, di ogni singolo componente del device:

Componente	Consumo Wake	Consumo DeepSleep
ESP32	50 mAh	0.05 mAh
RFNANO	19-30 mAh	0.05 mAh
Display OLED	20-35 mAh	0.9 mAh
MicroSD	40 mAh	5 mAh
RTC DS3231	0.8 mAh	0.08 mAh
	Totale	Totale
	~ 130-160 mAh	~7 mAh

Considerando una batteria al litio da 3.7V 500 mAh e considerando che i tempi di utilizzo del device e della media delle persone incontrate nell'arco di una giornata, otteniamo un consumo medio del device di circa 60 mAh (alternando le fasi di sonno profondo con quelle di accensione). Misurando i tempi di scarica della batteria otteniamo:

$$Durata\ Batteria = \frac{Capacità\ batteria\ mAh}{Corrente\ di\ carico\ mAh} = \frac{500}{60} = 8,33333 \approx 8\ ore\ e\ 20\ minuti$$

Aumentando la capacità della batteria potremmo portare il tutto a funzionare per un'intera giornata.

$$Durata\ Batteria = \frac{Capacità\ batteria\ mAh}{Corrente\ di\ carico\ mAh} = \frac{1000}{60} = 16,66666 \approx 16\ ore\ e\ 40\ minuti$$

Nel nostro caso, utilizzando i moduli adeguati al circuito finale otterremmo tali consumi:

Componente	Consumo Wake	Consumo DeepSleep
ESP32	50 mAh	0.05 mAh
Arduino Micro	5 mAh	0.05 mAh
NRF24L01+	10-20 mAh	0.04 mAh
Display OLED	20-35 mAh	0.9 mAh
MicroSD	40 mAh	5 mAh
RTC DS3231	0.8 mAh	0.08 mAh
	Totale	Totale
	~ 120-150 mAh	~7 mAh

Un consumo energetico di 10 mAh più basso rispetto a quello in fase di sviluppo.

5. APP SMARTPHONE E WEB APP

Il software serve per effettuare varie operazioni:

- Operazioni Utente:
 - Creare un accoppiamento con il proprio device → Individuo
 - Inserire/Modificare le proprie credenziali
 - Collegare device allo smartphone per scaricare i vari contatti avvenuti e inoltrarli al server
 - Controllare tutti i contatti giornalieri avvenuti e quelli totali
 - Mandare una segnalazione sul proprio stato di salute
 - Permettere di gestire il device e le varie comunicazioni
 - Gestire il trasferimento dei dati
- Opzioni Staff/Amministratori
 - Analisi dei dati raccolti per comprendere quali sono le aree geografiche più colpite dal virus
 - Raccolta delle segnalazioni al fine di effettuare tamponi mirati
 - Recupero delle identità degli individui a contatto con dei positivi
 - Invio repentino di informazioni a che è stato a contatto con positivi al fine di evitare l'espansione incontrollata del virus
 - Sarà possibile, inoltre, effettuare data mining sui dati ottenuti al fine di rilevare eventuali correlazioni fra lo stato di salute degli individui e il presentarsi del virus

Tale software è suddiviso in 3 parti fondamentali:

- Frontend:
 - Presenta la parte grafica da presentare all'utente
 - Gestito dal framework chiamato IONIC per lo sviluppo di Progressive Web App



- Backend:
 - Applicazione lato server adibita alla gestione dei dati provenienti dal frontend
 - Implementata in Express Js, la gestione delle richieste è eseguita attraverso delle REST GET, POST, PUT, DELETE che a loro volta eseguiranno delle operazioni CRUD su un DB MySQL

Express



- Database:
 - Storage dei dati
 - Gestito dall'RDBMS chiamato MySQL



5.1. Gestione dei dati

Come detto in precedenza per la gestione dei dati ci siamo affidati a MySQL, l'RDBMS per eccellenza, conosciuto in tutto il mondo per le sue ottime prestazioni, versatilità e semplicità di utilizzo.

Il nostro obiettivo sarà dunque quello di creare una base di dati in grado di contenere

- l'insieme degli utenti che faranno uso del nostro servizio e di quelli lo gestiranno
- l'insieme dei devices BubbleBox appartenenti agli utenti che faranno uso del nostro servizio
- l'insieme dei contatti avvenuti nel tempo fra gli utenti che faranno uso del nostro servizio
- l'insieme delle segnalazioni effettuate dagli utenti che faranno uso del nostro servizio

L'utente potrà dunque **registrarsi** al nostro servizio indicando alcuni dati personali come **nome**, **cognome**, **codice fiscale**, **data di nascita**, **indirizzo di residenza** e **sex**, inoltre verranno richiesti ulteriori informazioni significative come una **mail** e uno **username** per poter effettuare un accesso più agevole al servizio. Fatto ciò l'identità di un utente dovrà essere controllata tramite appositi strumenti, dunque verrà inviata una mail di conferma e il servizio sarà disponibile. Verranno associate all'utente ulteriori informazioni come il **ruolo** all'interno del servizio (user, staff, admin) e uno **stato di salute** che indicherà dunque le condizioni dell'utente ("sano", "con sintomi", "in quarantena", "positivo", "positivo asintomatico").

Una volta effettuata la registrazione, l'utente dovrà registrare il proprio device identificato nel sistema con un indirizzo mac. Parlando dunque dei devices, questi verranno rilasciati con un indirizzo mac (di natura univoco) prestabilito.

Dopodiché una volta associato il dispositivo l'utente lo potrà indossare per rilevare i contatti che avvengono fra di esso e gli altri individui che ne indossano uno. Ogni contatto comporta la ricezione del MAC relativo al dispositivo di un altro utente e la memorizzazione in locale di una stringa con il seguente format: **"MAC_del_mio_device|MAC_del_device_incontrato|timestamp"** che verrà successivamente inviata su un cloud online.

Quando un utente facente parte del servizio accusa i sintomi del virus, dovrà riportarli nell'apposita sezione adibita alle **segnalazioni**, i dati raccolti saranno dunque utili per comprendere le condizioni del segnalatore che sarà verrà prontamente soccorso. I dati da inserire saranno relativi ai **sintomi** tipici del virus in considerazioni. Effettuata la segnalazione lo stato dell'utente passerà da "sano" a "con sintomi", da questo punto in poi solo le autorità sanitarie potranno cambiare lo stato di salute dell'individuo; ad esempio, nel caso del COVID, poco dopo fatta la segnalazione, l'utente sarà soggetto a tampone e una volta conosciuto l'esito potrà passare o meno in stato di "positivo". Quando un individuo passa allo stato di "positivo", le autorità potranno entrare a conoscenza delle persone con le quali questo è stato a contatto definendo una finestra temporale congrua con i tempi di vita del virus. Le persone a contatto con quest'ultimo passeranno dunque in stato di "in quarantena" e verranno successivamente controllate dalle autorità sanitarie.

5.1.1. Schema ER

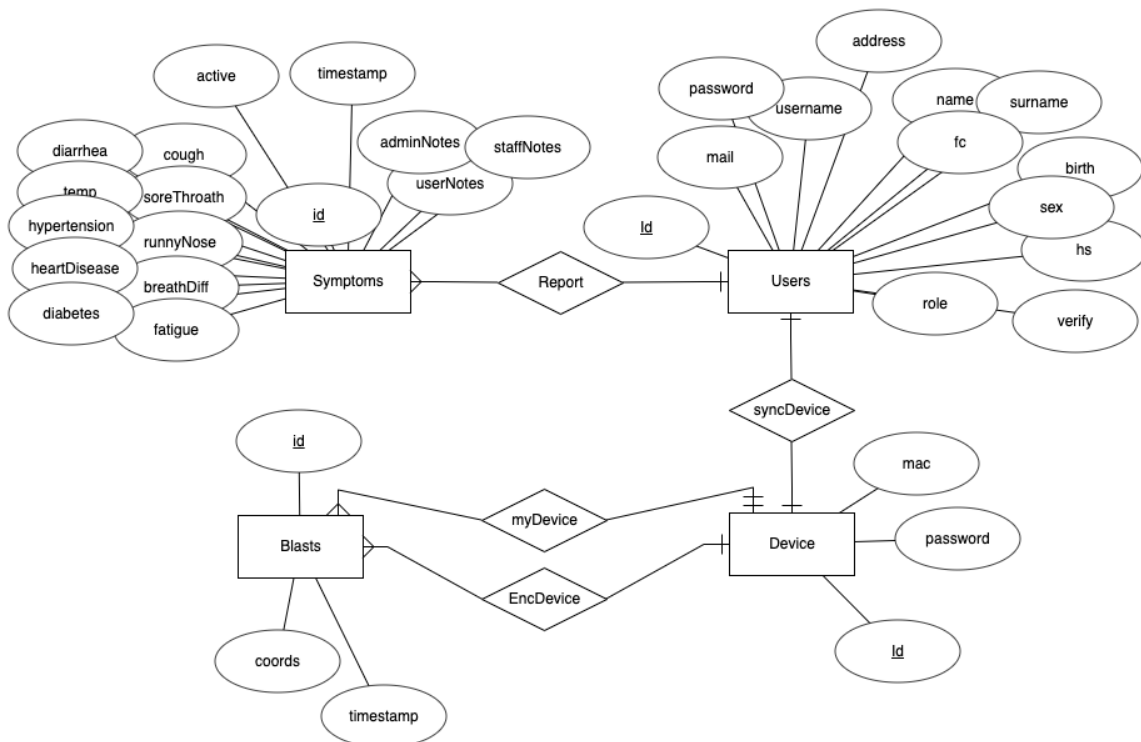
Dunque, una volta centrato l'obiettivo, identifichiamo quattro entità correlate tra di loro:

- Users, cioè l'insieme degli utenti che usufruiranno del servizio e quelli che lo gestiranno.
- Devices, cioè l'insieme di tutti i devices bubblebox.
- Blasts, cioè l'insieme di tutti i contatti che sono avvenuti nel tempo fra le istanze di Users
- Symptoms, cioè l'insieme di tutte le segnalazioni eseguite dagli utenti che hanno riscontrato dei sintomi tipici del Virus che si sta cercando di debellare (prenderemo per ora in esempio quelli relativi al COVID-19).

E possiamo inoltre individuare quattro relazioni che insorgono fra di esse:

- Una relazione tra Users e Devices, in quanto ogni utente avrà associato ad esso un device
- Una relazione tra Symptoms e Users in quanto ogni utente può eseguire delle segnalazioni
- Due relazioni tra Blasts e Devices in quanto nella prima verranno riportati i devices entrati in contatto e dunque ci saranno due attributi legati a Devices.

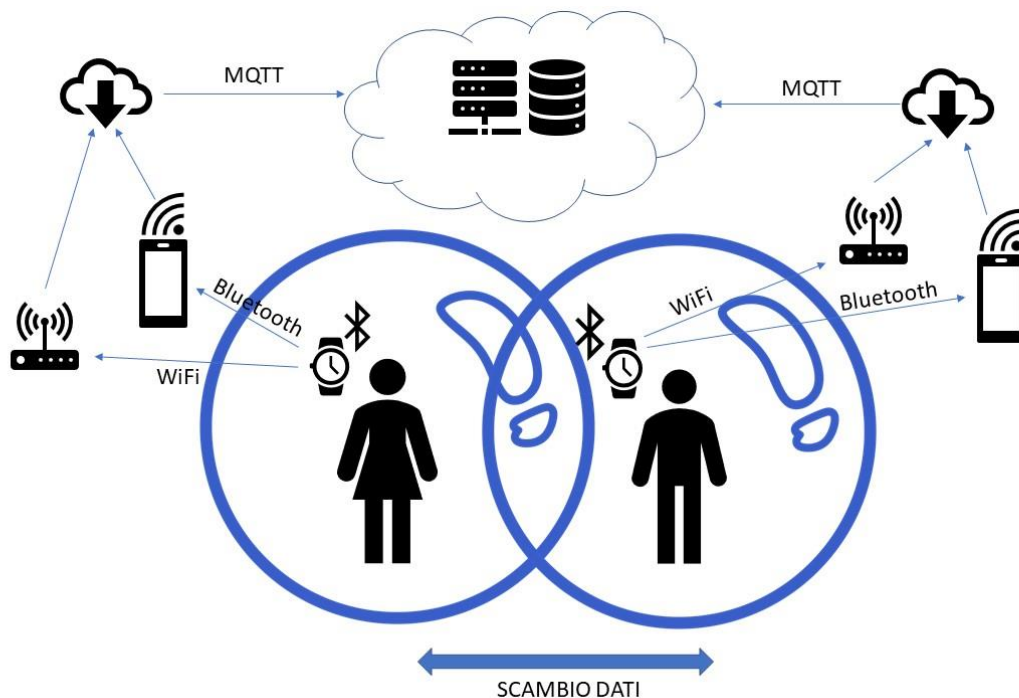
Ecco lo schema ER risultante:



Dallo schema ER estrapoliamo dunque il relativo schema relazionale:



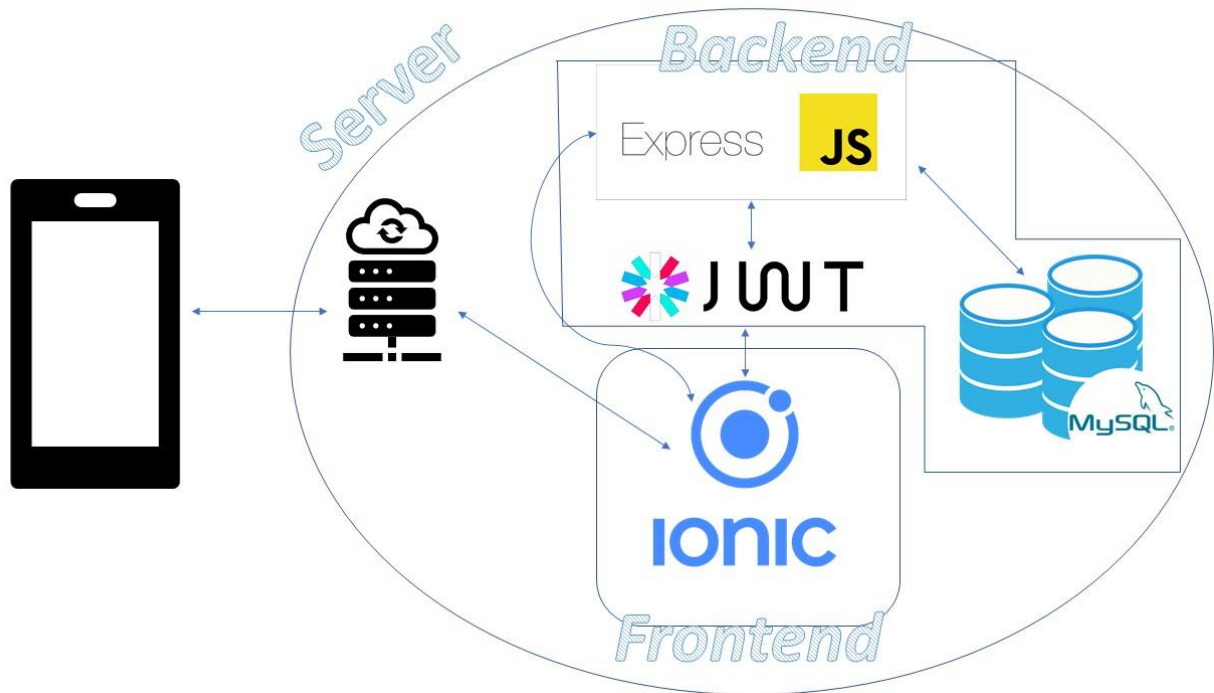
6. FUNZIONAMENTO DEVICE + APP



Come è possibile vedere dall'immagine, il funzionamento del device e da accorpate a quello dell'app smartphone o attraverso l'ausilio di altri mezzi di comunicazione, tra cui un Access Point (AP).

- Il device una volta raccolte le informazioni dei contatti le elabora ed effettua uno storage in locale:
 - Si dispongono di diversi file, uno per ogni giorno, in modo da avere un catalogo bene fornito e specifico di tutti i contatti avvenuti:
 - ESEMPIO:
 - Contact_2020_04_11
 - Contact_2020_04_12
 - ...
 - Contact_2020_05_30
 - ...
- Una volta raccolte le informazioni, queste possono essere mandate, quando si vuole, al server attraverso due canali di comunicazione che è possibile scegliere:
 - Bluetooth smartphone:
 - Collegandosi allo smartphone e passando in automatico tutti i contatti per poi essere inoltrati al server e messi in un DB
 - WiFi:
 - Collegandosi ad un AP attraverso un normalissimo protocollo WIFI o WPS:
 - WPS: collegarsi all'AP schiacciando il bottone presente sul router domestico e attendere la connessione
 - WIFI SSID PASS: collegandosi in modo tradizionale al WiFi di casa inserendo le credenziali di accesso che verranno passate quando ci si collega allo smartphone (visto che il device non ha un tastierino con cui comporre messaggi)

7. FUNZIONAMENTO APP



Il funzionamento della App è un po' più articolato del resto, perché ogni richiesta che passa al server deve essere prima inoltrata al Backend, verificata e poi eseguita. Questo meccanismo di sicurezza vieta a chiunque di eseguire delle chiamate se non si dispongono dei permessi adeguati.

Il workflow è il seguente:

- Lo smartphone si connette al servizio attraverso l'apposita App (Android e IOS)
- L'utente avrà un proprio username e password per accedervi
- Quando l'utente effettua delle richieste: aggiunte, modifiche ecc... disporrà di un certo token generato attraverso il processo di generazione di token di JWT, quindi ogni volta che un client accede avrà un token sempre diverso e nessun'altro potrà effettuare le sue richieste, ma solo lui e lui solo
- Una volta che il client esegue un'operazione, questa passerà prima dal backend, che controllerà se tale azione potrà essere eseguita e poi effettuerà le dovute manovre, come salvare i dati sul DB o modificarli
- Tutte le richieste che vengono effettuate hanno controlli in ogni parte del software, dal frontend al backend e perfino sul DB, attraverso dei trigger e controlli preliminari di accesso e gestione
- Una volta che la richiesta è stata effettuata verrà restituita l'azione specifica all'App, in modo da dare un messaggio visivo e chiaro all'utente che sta utilizzando l'applicazione

TECNOLOGIE DEVICE

Le tecnologie utilizzate nella realizzazione del device e dell'app sono diverse:

- **Bluetooth:**

- La tecnologia Bluetooth è sicuramente la tecnologia più importate che viene utilizzata in questo progetto, in primis perché serve ad intercettare i contatti che avvengono con le persone ma anche per la comunicazione e il passaggio di dati attraverso lo smartphone.



- La tecnologia Bluetooth ha tante varianti e quella che si è utilizzata è chiamata BLE (Bluetooth low energy), cioè un Bluetooth con un basso utilizzo di energia.



- **WiFi:**

- La tecnologia WiFi viene utilizzata per permettere al device di connettersi, ad una rete domestica, che disponga di un WiFi, per poter inoltrare i dati al server con tutti i contatti avvenuti.



- La connessione al WiFi avviene attraverso uno scambio di informazioni con lo smartphone in cui l'utente dare al device le informazioni necessarie per connettersi ad una rete (SSID, PASS)
- La connessione al WiFi può avvenire anche in un altro modo, attraverso il metodo WPS, in cui l'utente, schiacciando il bottone con la sigla di quest'ultimo, può attivarlo sul proprio router di casa.

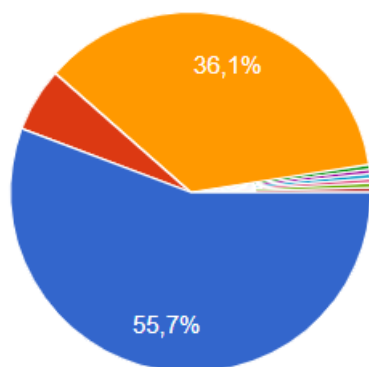


RACCOLTA DATI (Questionario)

Durante la fase progettuale e durante i vari test, per verificare le varie tecnologie da utilizzare, si è somministrato un questionario, totalmente anonimo, attraverso diversi canali comunicativi, per sapere cosa ne pensava la gente su un particolare device o App per gestire l'emergenza.

Questi sono i dati che si sono tirati fuori da 250 risposte:

- Per prima cosa è stato chiesto cosa ne pensava la gente se ci fosse stato un "qualcosa" che avrebbe potuto gestire e aiutarci contro l'emergenza COVID-19 accedendo ad alcuni dati sensibili di ogni singola persona; queste sono state le risposte:



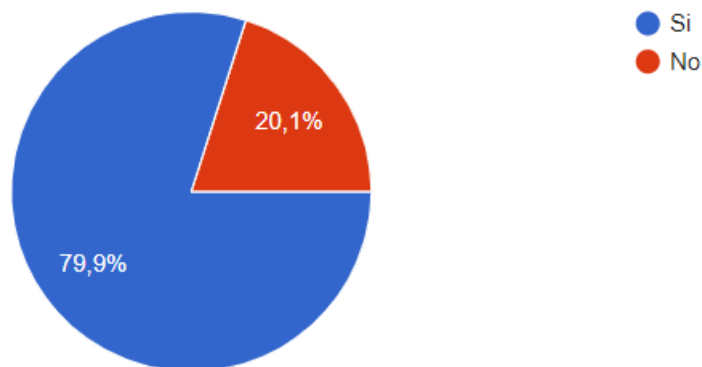
il 36,1% ha dato la sua negazione a questa domanda
più del 55% ha acconsentito a questo
il restante 9% (circa) si è astenuto oppure ha rivolto delle lamentele a questa richiesta.

già questo fa emergere un dato abbastanza significativo sul modo di pensare delle persone, anche se la maggior parte fosse d'accordo, ci sarebbe comunque una parte, una bella fetta di popolazione che non vuole acconsentire a tutto

questo e non far gestire i propri dati personali a discapito di tutto.

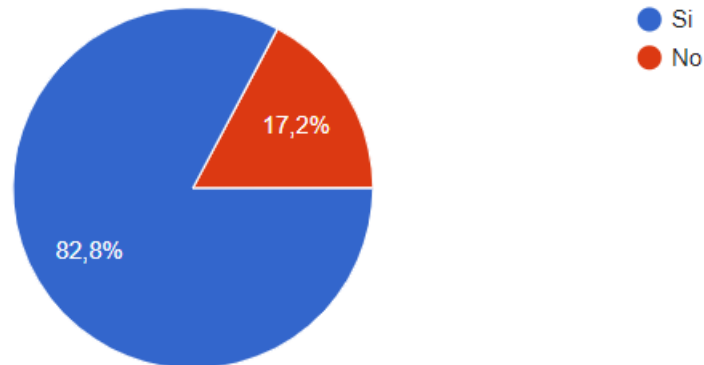
In totale possiamo dire che 56% ha detto sì e 44% ha detto no, e questo va contro l'utilizzo dell'App "Immuni" per poter essere efficace (>60%).

- A quelle persone che hanno risposto di no, è stato chiesto il perché di questa risposta. Le risposte sono state delle più svariate, ma comunque rimane il non consenso da parte di quasi metà dei test.
- La successiva domanda è stata rivolta a tutti (sia sì che no) e riguarda un argomento molto sentito negli ultimi tempi:
 - Pensi che nei tempi in cui viviamo siamo costantemente "spiati"?



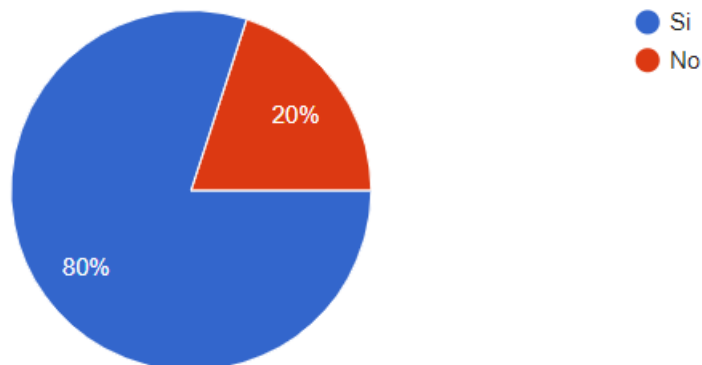
Qui quasi l'80% delle persone ha confermato la domanda. Quindi anche quelli che avevano risposto sì all'utilizzo di alcuni dati sensibili sa che, comunque sia, viviamo in un modo dove siamo perennemente spiati.

- Le domande che arrivano subito dopo, parlano di social e di utilizzo di quest'ultimi. La serie di domande viene proposta per arrivare a quella più fantomatica, la seguente:
 - Se ci dovesse essere una App o un dispositivo in grado di fronteggiare ogni virus possibile, ma che userebbe i tuoi dati personali, lo useresti?



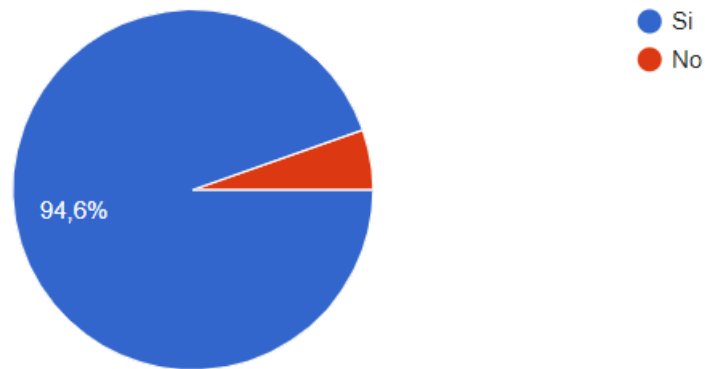
Qui la risposta della maggior parte delle persone è positiva, un po' in contrasto con la primissima domanda. Quindi, le persone, una volta messe in un contesto in cui viene fatto capire che forse la App è l'ultimo dei nostri problemi di privacy, ha cambiato la propria idea, trasformandola in positiva. Questo fa ben sperare sui futuri sviluppi del device o di una qualche APP.

- Un'altra delle domande che è stata chiesta è abbastanza interessante, perché è riferita solamente a quelle persone che hanno risposto in modo negativo alla precedente:
 - Se il dispositivo utilizza i dati in modo trasparente e chiaro all'utente finale, saresti d'accordo?



L'80% delle persone che ha risposto in modo negativo alla domanda precedente ha fatto marcia indietro ed ha deciso di adottare la soluzione, anche se questa utilizza una parte di dati sensibili.

- La domanda successiva chiede, più nello specifico, come i dati della persona venissero trattati:
 - Vorresti che al dispositivo fosse associata anche una app per lo smartphone in grado di monitorare tutte le informazioni che sono state raccolte, in modo che tutte queste informazioni siano gestite attraverso un meccanismo del tutto chiaro e trasparente?



Quasi in totalità, le persone hanno dato una risposta positiva. Quindi sì al device (o APP) ma che si gestisca al meglio i dati sensibili e che il tutto avvenga in modo trasparente per ogni individuo.

Quello raccolto da questo questionario è stato il nostro punto di partenza della struttura, la gestione e la utilizzabilità del device, della APP e di tutti i dati che dovranno essere gestiti e controllati in modo sicuro e trasparente all'individuo.

CONCLUSIONI

Nell'ultimo periodo sono tante le applicazioni e i device che sono stati proposti per fronteggiare l'emergenza "coronavirus", parecchie delle quali utilizzano solamente un'analisi dei dati, partendo da dati già raccolti, ma non sempre accurati, altre utilizzano smartphone attraverso la geolocalizzazione, ma inefficienti in ambienti chiusi o schermati.

Alcune software house hanno proposto delle App per smartphone in grado di raccogliere i dati con la tecnologia Bluetooth, ma, per il momento, sono poche le notizie che specificano in che modo verranno raccolti i dati e se ci saranno delle limitazioni a livello di sistema operativo.

Dal nostro punto di vista, quando si è iniziato a muovere i primi passi nel progetto, si è pensato di utilizzare una semplice app smartphone, in modo che quest'ultima potesse essere facilmente realizzabile, con un costo irrisorio e con una distribuzione molto più elevata, ma si sono verificate diverse problematiche dovute alla struttura stessa dei sistemi operativi smartphone.

- *La prima problematica* la si è riscontrata su IOS, che mette a disposizione un servizio Bluetooth che filtra ed *elimina tutti device Bluetooth non certificati MFI*;
- *La seconda problematica* riscontrata (sempre su IOS) si è avuta sulla ricerca dei dispositivi Bluetooth in zona, infatti non riuscivamo a vedere il MAC Address dei device ma *solamente l'UUID* e questo ci ha creato delle problematiche alla realizzazione dell'idea e all'implementazione stessa del codice di sviluppo (UUID cambia sempre e spesso e non è un ID da poter considerare per identificare in modo univoco un device);
- *La terza problematica*, questa volta, è arrivata da Android, infatti, potevamo visualizzare tutti i dispositivi Bluetooth con i loro MAC Address, ma non riuscivamo ad accedere al nostro MAC privato per motivi di sicurezza imposti da Android (avevamo bisogno di conoscere il nostro MAC per utilizzarlo come ID ed essere identificato in modo univoco, nel sistema, come *"persona in possesso di un device BubbleBox"*).

Tuttavia, si sono riscontrate anche altre problematiche nell'utilizzo dello smartphone:

- Una problematica, non di poco conto, era sicuramente data dal consumo eccessivo di batteria provocava dallo scanning continuo del Bluetooth;
- Un'altra è il non possesso di alcun tipo di smartphone o il possesso di telefoni datati non adatti al compito:
 - Anziani, bambini, senzatetto, persone che non ne vogliono uno...
- Problematica dovuta alla poca, o scarsa, consapevolezza nell'utilizzo di determinate tecnologie:
 - Accensione del Bluetooth, avvio dell'App, controllo applicazione...
- E per finire il sovraccarico del server alle continue richieste che avrebbe ricevuto dall'applicazione.

Prima di arrivare a queste conclusioni, grazie ai test effettuati, la nostra idea, in principio, era strutturata in questo modo:

- Ogni smartphone faceva parte del sistema BubbleBox attraverso un'iscrizione all'app (tramite mail e password);
- Ogni smartphone era identificato in modo univoco attraverso il MAC Address (unico per ogni device);
- Quando si accedeva all'app, automaticamente si accendeva il Bluetooth e partiva la scansione periodica dell'area alla ricerca di altri device Bluetooth;

- I device, appartenenti alle persone, dovevano essere identificati rispetto agli altri milioni di device che sono in giro (cuffiette, Smart band, smartwatch e altro) e questo ha creato un altro problema non di poco conto che poteva essere gestito in questo modo:
 - Ogni volta che si trovava in zona un Bluetooth ad una certa distanza (considerando l’RSSI che misura la potenza del segnale) lo si considerava e partiva una POST al server per poter aggiungere tale contatto in una tabella dei “contatti avvenuti” (chiamata *blast* per via dell’onomatopea impura dello scoppio della bolla);
 - Questo identificativo (MAC Address) incontrato, veniva confrontato, lato server, con la tabella dei device presenti nel sistema, per vedere se tale ID appartenga ad una persona inserita nel sistema;
 - Se il riscontro fosse stato positivo, si sarebbe aggiunto il contatto all’interno della tabella del DB, altrimenti tale ID del device veniva inserito, in locale, sullo smartphone della persona da cui è partita la segnalazione, in modo che ogni volta che avrebbe incontrato quel device avrebbe saputo già a prescindere che apparteneva ad un dispositivo che non faceva parte del sistema e quindi avrebbe evitato controlli inutili al server.

In questo workflow e nelle problematiche esposte in precedenza ci hanno fatto rivalutare l’uso dello smartphone. Infatti, si è pensato di passare il tutto ad una gestione più semplice, efficiente e facilmente indossabile da chiunque senza obbligo di smartphone: un device dalle sembianze di un orologio da polso (smartwatch).

La nostra soluzione di device copre tutte le pecche e le imperfezioni di altre applicazioni, idee e problematiche esposte in precedenza, in modo tale da avere sempre una raccolta precisa dei dati in ogni luogo e modo, al fine di poter essere utilizzati e analizzati.

Come tutte le considerazioni fatte, anche il device ai suoi pro e contro, che verranno descritti di seguito:

- PRO:
 - Leggero e maneggevole
 - Poco costoso
 - Implementazione agevole
 - Batteria dedicata
 - Bassi consumi
 - Indossabile da chiunque
- CONTRO:
 - Tempi di realizzazione corposi
 - Tempi di distribuzione lunghi ed onerosi

Sicuramente nella creazione del device abbiamo molti aspetti piacevoli ma anche alcuni negativi che bisogna tenere conto. Sicuramente una App ha una velocità di distribuzione elevata perché un software riesce a girare molto più velocemente e coprire molte più persone a livello locale, nazionale e mondiale. Tuttavia, anche la realizzazione tramite App ha delle pecche molto forti, alcune delle quali ampiamente discusse precedentemente, infatti, alcune software house che si stanno occupando di creare un’apposita applicazione per gestire l’emergenza, in questo momento (fine Aprile 2020), stanno pensando di utilizzare un device per coprire una parte della popolazione italiana, perché la App ha senso di esistere se almeno il 60% (e poco più) della popolazione la usa in modo impeccabile e in ogni momento. Ritornando al dato di prima, ricordiamo che l’Italia è il paese più longevo d’Europa (forse del mondo) con ben più del 22% della popolazione over 70. Questo dato è abbastanza chiaro se si pensa che buona parte di questa percentuale non possiede o non sa cosa sia uno smartphone. A questo c’è da aggiungere sicuramente una percentuale di almeno un 15% dei bambini, che non possiedono uno smartphone e a tutte quelle persone che non ne vogliono possedere uno. Anche se lo stato, imponesse per legge, che tutti devono tenere,

obbligatoriamente, uno smartphone con uno strumento di tracciabilità continua (con grande consumo di batteria) sempre acceso, questo dovrebbe prevedere un acquisto forzato e obbligatorio di uno smartphone anche per i più neofiti.

Il nostro sistema copre questa parte della popolazione e tutto il resto, perché fornisce un device dove la tua unica preoccupazione sarà quella di metterlo in carica (una volta al giorno o ogni due giorni (forse) la sera) contro il fatto di avere un telefono sempre disponibile, acceso e con un consumo di batteria elevato.

Sicuramente nelle ultime settimane vedremo come si evolverà la situazione, la nostra idea rimane, tuttora, un'idea innovativa e valida che può essere utilizzata ed inglobata, insieme altre proposte presente in questo momento sul settore, ma anche in tantissimi altri ambiti.

MANUALE D'USO

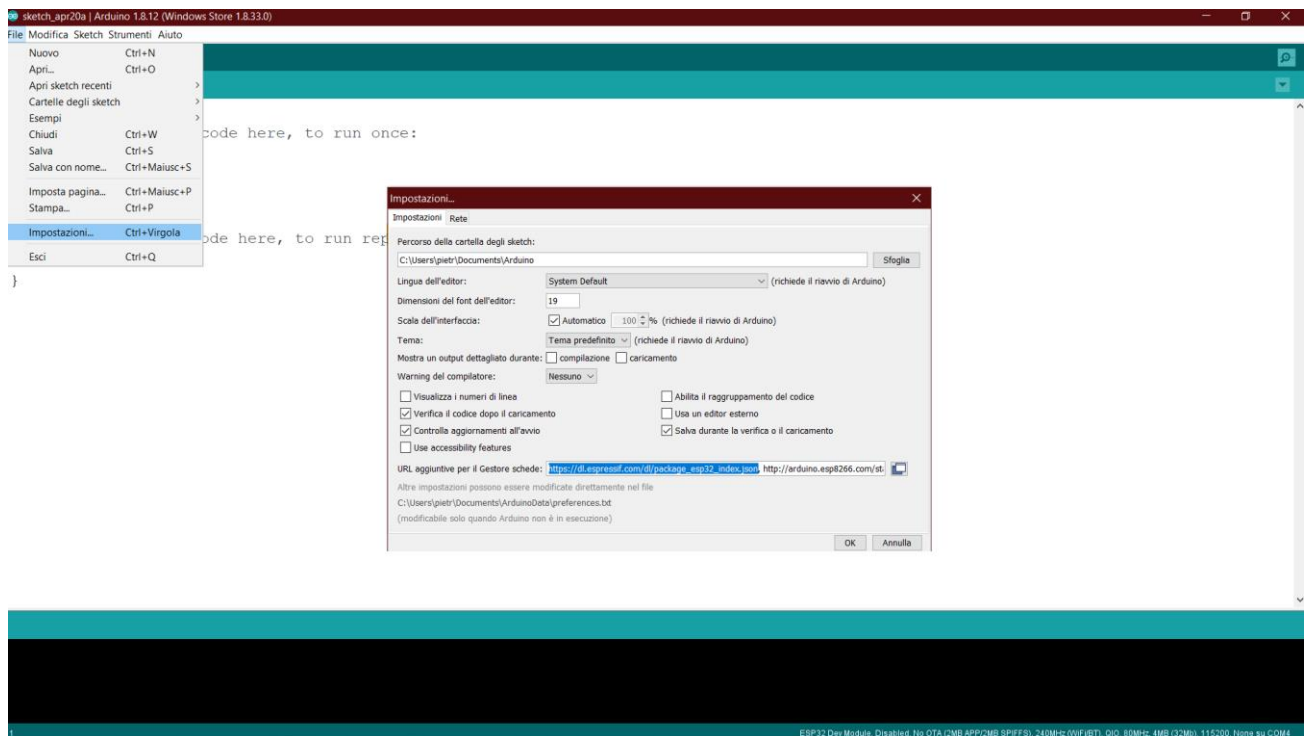
1. INIZIALIZZAZIONE DEVICE

Per poter inizializzare il device bisogna seguire i seguenti passi:

- Installare la IDE di Arduino sulla pagina ufficiale: <https://www.arduino.cc/en/Main/Software>

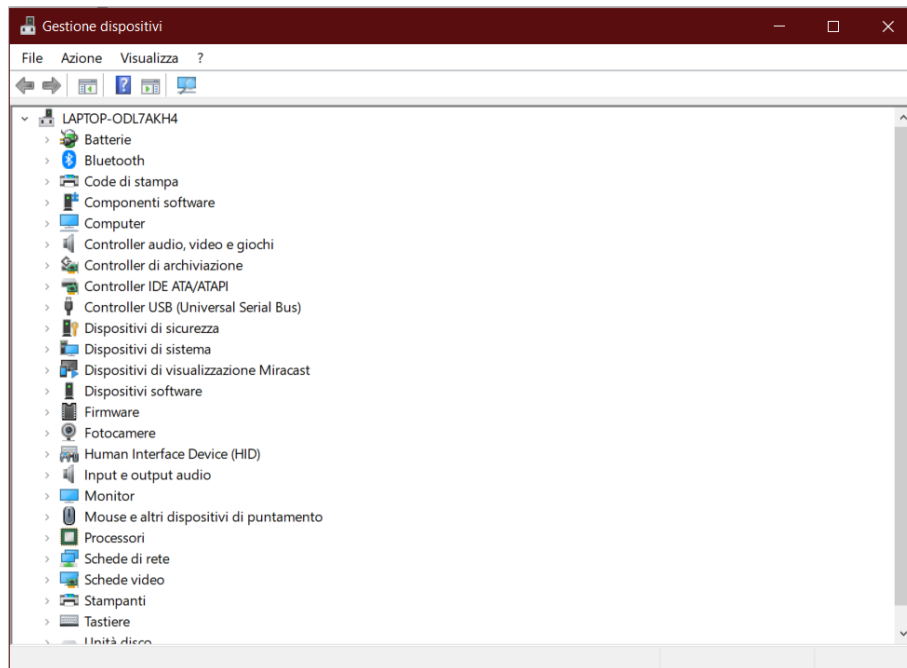


- Installare CORE ESP32 sulla IDE di Arduino:
 - Arduino IDE > File > Impostazioni > Schede > URL Schede > https://dl.espressif.com/dl/package_esp32_index.json



- Arduino IDE > Strumenti > Scheda > ESP32 Dev Model
 - Arduino IDE > Strumenti > Partition Scheme > NO OTA (2MB APP/2MB SPIFFS)
- Installazione librerie:
 - Installazione librerie ESP32: <https://github.com/espressif/arduino-esp32>
 - Installazione librerie RFNANO: <https://github.com/maniacbug/RF24>
- Possibili problematiche:
 - è possibile che quando si connette l'ESP32 alla porta USB del PC non venga letta
 - aprire Gestione Dispositivi
 - vedere se tra i device presenti c'è qualche problema o warning

- andare sul problema e cliccare col tasto destro > **Aggiorna Driver**
- il problema dovrebbe essere risolto riavviando la IDE di Arduino



2. FIRMWARE SCHEDE


- Clonare la repo
 - `git clone` <https://orteip94@bitbucket.org/bbteam/bubblebox-device.git>












/			
Name	Size	Last commit	Message
Business Plan e Presentazione		4 hours ago	Modifica schema e immagini device e funzioname...
Circuito		17 hours ago	Modify Image Circuits
Library		2020-04-10	add library BLT serial
Logo		2020-04-05	add library
Schema		4 hours ago	Modifica schema e immagini device e funzioname...
Sketch Device		15 hours ago	Bug FIX conn BLT Serial OK
README.md	7.5 KB	15 hours ago	README.md edited online with Bitbucket

- Installare gli sketch sulle schede:
 - RFNANO:
 - Aprire Arduino IDE
 - Inserire la scheda RFNANO
 - Arduino IDE > Strumenti > Scheda > Arduino Nano > Processore:ATMega328P
 - Selezionare Porta
 - Installare sketch contenuto nella directory Sketch Device/RFNano

bubblebox-device / Sketch Device / RFNano			
Name	Size	Last commit	Message
..			
RFNano.ino	3.77 KB	18 hours ago	Bug Fix RF NANO

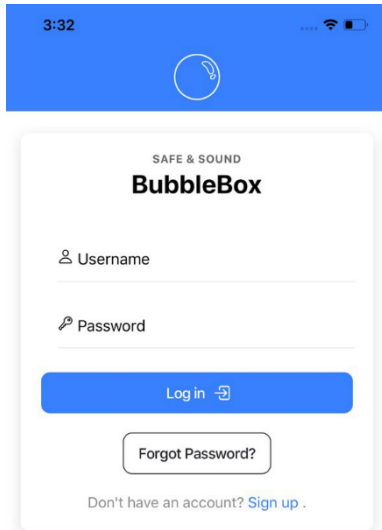
- ESP32:
 - Aprire Arduino IDE
 - Inserire la scheda ESP32
 - Arduino IDE > Strumenti > Scheda > ESP32 Dev Model
 - Arduino IDE > Strumenti > Partition Scheme > NO OTA (2MB APP/2MB SPIFFS)
 - Selezionare Porta
 - Installare sketch contenuto nella directory Sketch Device/ESP32/Device_Sleep&Wake

 bubblebox-device / Sketch Device / ESP32 / Device_Sleep&Wake / **BubbleBox_Device**

Name	Size	Last commit	Message
 ..			
 BubbleBox_Device.ino	7.85 KB	15 hours ago	Bug FIX conn BLT Serial OK
 Display_GestioneDisplay.h	7.41 KB	15 hours ago	Bug FIX conn BLT Serial OK
 Display_IconeDisplay.h	19.73 KB	3 days ago	connect BUBBLESTATION and send data server OK
 ESP32_BLEDevice.h	5.41 KB	5 days ago	WPS connection Bug Fix
 ESP32_SerialBLT.h	4.07 KB	15 hours ago	Bug FIX conn BLT Serial OK
 ESP32_SleepWake.h	2.01 KB	2020-04-10	Change name directory project
 ESP32_WiFi.h	14.72 KB	yesterday	Add connection Multi WIFI
 RTC_Clock.h	2.47 KB	3 days ago	Set Data Time RTC when ESP32 connect WIFI
 SD_GestioneSDCard.h	7.41 KB	yesterday	Add connection Multi WIFI
 SelectFunction_Button.h	1.08 KB	3 days ago	Set Data Time RTC when ESP32 connect WIFI

3. APP

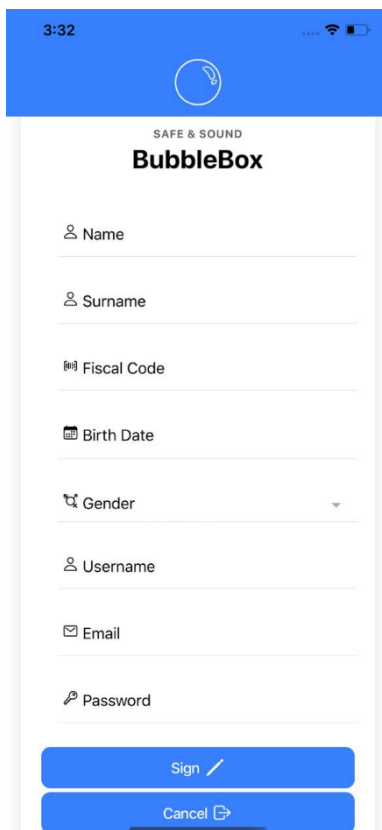
La schermata iniziale dell'app è molto semplice, infatti, verrà chiesto di entrarci, con le proprie credenziali, oppure registrarsi, se non si dispone di quest'ultime.



Nel campo "Username" verrà inserito il nome utente che si è creato in fase di registrazione, mentre la password sarà creata anch'essa in fase di registrazione.

Se si è dimenticati la password sarà possibile cliccare su "Forgot Password?" per recuperarla attraverso i vari metodi di recupero.

La schermata registrazione è la seguente:

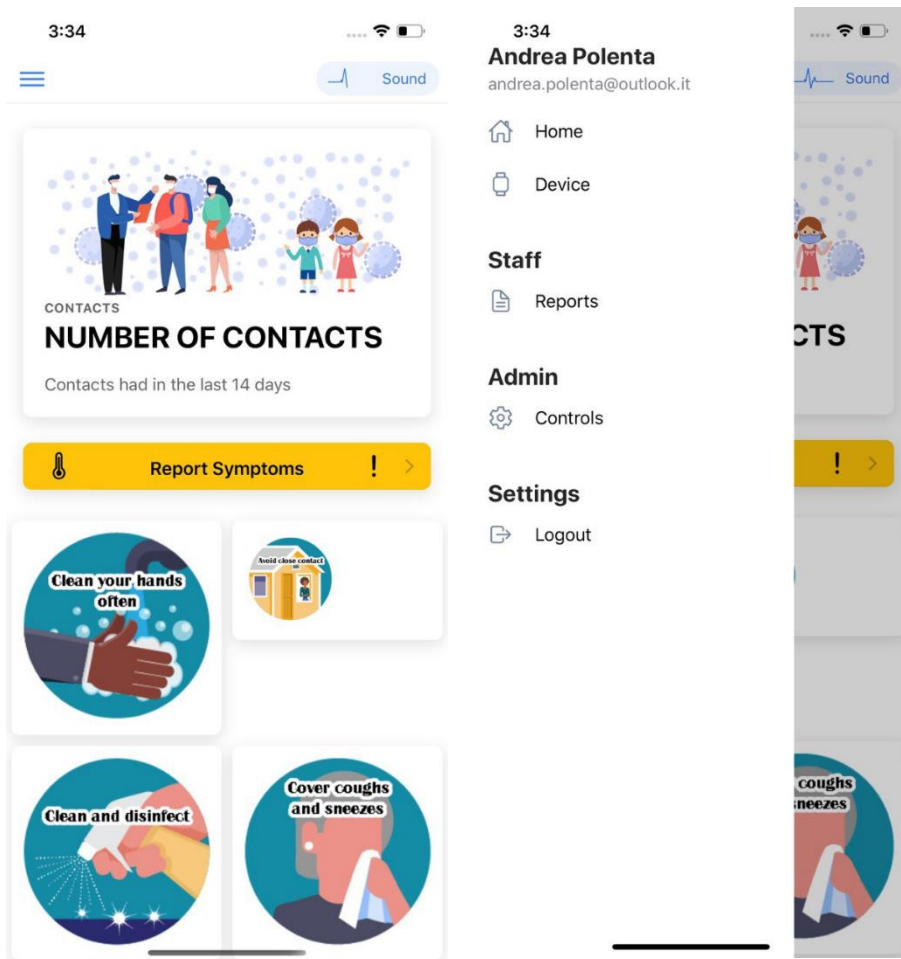


La schermata presenta vari campi da compilare per potersi registrare ed entrare per gestire le varie opzioni e il device di cui si dispone

I campi sono i seguenti:

- Name:
 - a. Nome della persona che si vuole registrare
- Surname:
 - a. Cognome della persona che si vuole registrare
- Fiscal Code:
 - a. Codice fiscale
- Birth Date:
 - a. Data di nascita
- Gender:
 - a. Sesso
- Username:
 - a. Nome utente con il quale accedere all'app per entrare nella propria area riservata
- Email:
 - a. Mail dove ricevere le varie notifiche e recupero password
- Password:
 - a. Password con cui si accederà al servizio

Schermata di home di accesso:



La schermata di home presenta un menù in alto a sinistra che una volta cliccato ci aprirà una sequenza di azioni da poter eseguire:

- Home: home app
- Device: inserimento e gestione device

Le altre impostazioni sono riservate solo allo staff e admin per la gestione delle emergenze e segnalazioni.

Nella schermata home sarà presente il numero dei contatti giornalieri e quelli degli ultimi 14 giorni, ma non si potrà sapere con chi si è venuto a contatto (per privacy).

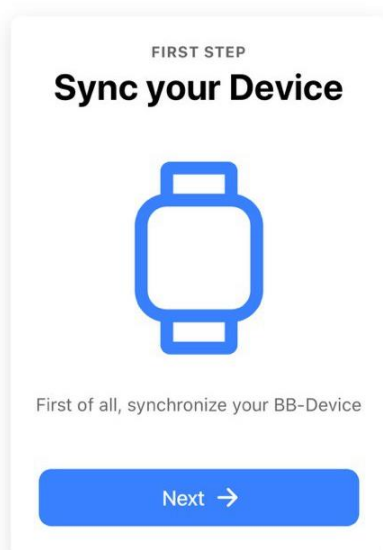
Sarà presente un bottone "Report Symptoms" in cui sarà possibile effettuare una segnalazione dei propri

sintomi, qualora si pensi di essere infetti.

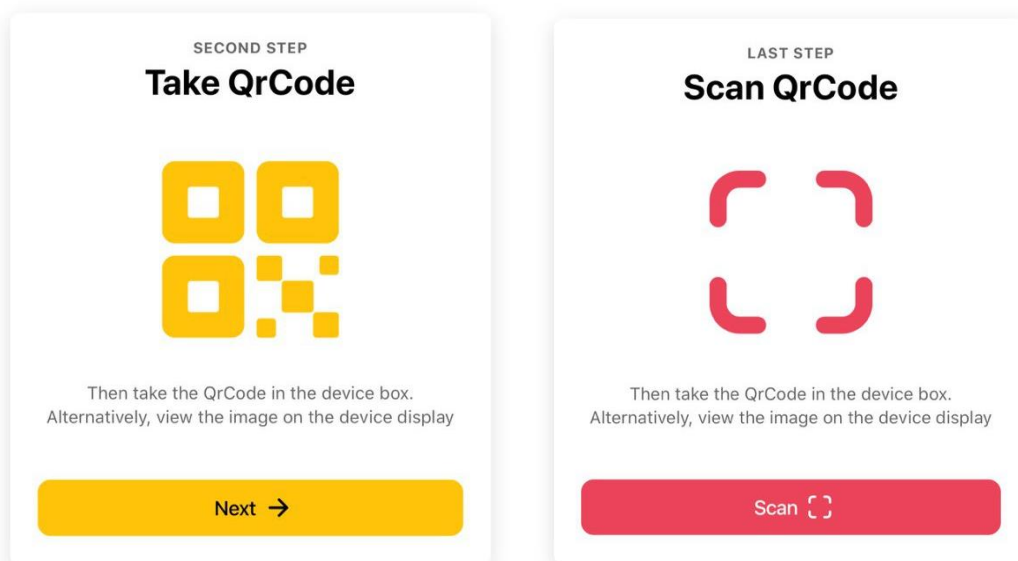
Saranno presente, in fondo alla schermata, delle buone regole da seguire e rispettare.

In alto a destra della schermata sarà presente lo stato di salute attuale.

Cliccando su "Device" presente nella home, potremo gestire il device:



Se non si è inserito alcun device sul proprio profilo, verrà richiesto di inserirne uno. Seguendo i vari passaggi visualizzeremo le seguenti schermate:



Verrà chiesto di scansionare il QRCode del dispositivo BubbleBox, in modo da inserirlo tra i propri dispositivi. Il QRCode sarà presente nella confezione del device e nel display andando nell'apposita sezione. Basta inquadrare tale codice e il device verrà inserito.

4. MOCKUP DISPLAY DEVICE

