

Progetto programmazione mobile

Specifiche dell'applicazione

L'applicazione deve automatizzare l'apertura di un varco tramite un codice QR. Il codice QR è caratterizzato da una posizione GPS; quando l'utente si avvicina a tale codice, e la scansione viene effettuata con successo, avviene l'apertura del varco. La risposta di avvenuta apertura o no viene segnalata all'utente tramite una notifica. Si deve avere un log degli accessi autorizzati e non autorizzati

Informazioni generali sull'applicazione

L'applicazione SmartKeys permette di gestire l'apertura di un varco in maniera automatica. Ogni varco è associato ad una chiave SmartKey. Scansionando la chiave con uno smartphone che si trova nei paraggi di questa, l'applicazione aprirà il varco associato. Ogni scansione termina con un messaggio di apertura o di non apertura varco che viene inviato tramite notifica. Tutte le scansioni effettuate si trovano in un'apposita schermata di cronologia scansioni, per ognuna delle voci presenti, verrà inserita anche la data e ora di quando è stata effettuata la scansione.

Informazioni tecniche sull'applicazione

I codici QR all'interno dell'applicazione prendono il nome di SmartKey. Tutte le SmartKey vengono registrate sul database Cloud Firestore di Google e sono caratterizzate da:

1. Un campo stringa **smartKeyCode** dove troviamo il codice QR sotto forma di stringa. Questo codice deve essere univoco.
2. Un campo Geopoint **smartKeyLocation** dove vengono inseriti latitudine e longitudine del luogo di posizionamento del codice.
3. Un campo stringa **email** che associa una SmartKey al corrispondente account SmartKeys
4. Un campo stringa **smartKeyName** dove viene registrato il nome della SmartKey.
5. Un campo stringa **smartKeyStreetAddress** dove viene messo l'indirizzo della SmartKey.

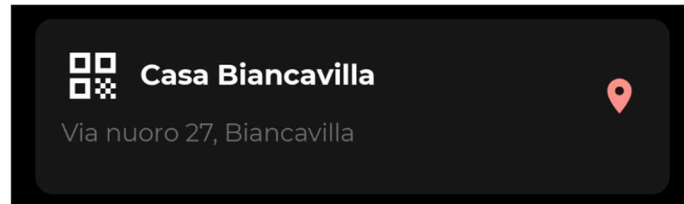
Se i primi tre campi sono necessari per il buon funzionamento dell'applicazione, i restanti due vengono aggiunti solo per dare qualche informazione in più all'utente dall'interfaccia dell'app. Gli ultimi due campi, perciò, possono essere definiti opzionali e non necessari per la corretta apertura del varco.

```
email: "patricknicolosi99@gmail.com"  
smartKeyCode: "ciao1234"  
smartKeyLocation: [37.638695386266° N, 14.869615099067035° E]  
smartKeyName: "Casa Biancavilla"  
smartKeyStreetAddress: "Via nuoro 27, Biancavilla"
```

Schermata catturata dalla console di Firebase.

L'applicazione all'avvio permette di effettuare il login, o di registrarsi come nuovo utente. Tutto questo viene implementato sfruttando il servizio FirebaseAuth. L'applicazione è per cui pensata, per gestire più utenti, dove ogni utente visualizzerà nella propria applicazione solo ed esclusivamente le proprie SmartKey. Per il buon funzionamento di questo meccanismo di visualizzazione ecco che risulta fondamentale il campo **email** su Cloud Firestore visto precedentemente. Infatti, l'applicazione filtrerà in una ListView solo le SmartKey che hanno come corrispondenza l'email tra il campo su Firestore e l'utente loggato sull'app.

Gli altri due campi **smartKeyName** e **smartKeyStreetAddress** tornano utili per dare qualche informazione in più all'utente che visualizzerà questi campi come segue (In bianco smartKeyName, in grigio smartKeyStreetAddress).



Ogni elemento della lista ha anche un ImageButton a destra che ci permette di visualizzare la SmartKey su Google Maps tramite l'avvio di un Intent a cui passiamo latitudine e longitudine della SmartKey.

È possibile scansionare una SmartKey dall'apposito pulsante "Scan a SmartKey". Se la geolocalizzazione sul dispositivo risulta disabilitata, verrà visualizzato un Dialog che ci informa di questo e che permette di navigare rapidamente alla sezione "Servizi di localizzazione" delle impostazioni e abilitare tale funzione. Se la localizzazione è attiva ed ha ottenuto un risultato (è stato localizzato il dispositivo), allora verrà avviata l'activity per scansionare il codice Qr. Per scansionare il codice è stata utilizzata la libreria Zxing. Una volta scansionato la SmartKey, verrà fatto prima un controllo che ne verifica l'esistenza andando a confrontare il risultato della scansione con i dati presenti su Firestore e successivamente verrà utilizzata la posizione del dispositivo per capire se tale SmartKey è stata scansionata da uno smartphone che si trova nei paraggi. Se la scansione viene effettuata mantenendo il telefono nell'arco di 50 m dalla SmartKey, allora verrà aperto il varco, altrimenti verrà restituito un messaggio di errore sempre tramite notifica.

L'avviso dell'avvenuta o non apertura di un varco sfrutta il servizio Firebase Cloud Messaging. Verrà visualizzata per cui una notifica che darà delle informazioni sul risultato di una scansione. Per interfacciare l'applicazione con l'invio di un messaggio su Firebase Cloud Messaging è stato scritto un breve script in NodeJS per il server. Verrà fatta una richiesta http al server tramite Volley, passando tramite parametri il token del dispositivo a cui inviare la notifica e il testo che deve contenere la notifica. Il token che passiamo tramite parametri al link <https://chrome-flawless-roadway.glitch.me/?token=myToken¬ificationBody=Hello> verrà "catturato" al primo avvio dell'app e salvato nelle SharedPreferences. Una volta terminata una scansione, verrà registrato il risultato di tale scansione in una cronologia. L'applicazione presenta per cui una lista delle scansioni effettuate con successo o no. Ogni elemento della lista è caratterizzato oltre che dalle informazioni di base, anche dalla data e ora della scansione.

Ricerca e correzione dei bug

1. Il bottone per scansionare le SmartKey viene visualizzato solo se è presente almeno una SmartKey sulla schermata principale. Facendo così impediamo dei confronti superflui sulle scansioni effettuate. Inoltre, facendo dei confronti su dei dati potenzialmente non presenti, riduciamo il numero di crash dell'applicazione. In questo modo possiamo evitare anche di controllare la presenza o meno di una connessione dati attiva. Se la connessione dati non è attiva, infatti, non viene caricata nessuna SmartKey e se nessuna SmartKey viene caricata il bottone non viene visualizzato. Se la connessione dati è attiva e vengono scaricate da Firestore le SmartKey, allora può essere effettuata la scansione. La scansione in questo secondo caso può essere fatta anche se la connessione viene interrotta durante, infatti, i dati sono ormai presenti in cache.
2. Se il dispositivo è in attesa di essere localizzato, verrà ricevuto un messaggio tramite Toast e non verrà avviata l'Activity di scansione del Qr. Se infatti scansioniamo una SmartKey, verrà fatto un controllo sulla vicinanza o lontananza del dispositivo, ma se il dispositivo non fosse stato localizzato, questo controllo potrebbe causare crash.

3. Avviare una scansione solo se la geolocalizzazione è attiva sul dispositivo. Se la geolocalizzazione non fosse attiva infatti verrebbe invocato un metodo che sfrutta un sensore al momento inattivo, con probabilità di crash.

Permessi Manifest

- ACCESS_FINE_LOCATION e ACCESS_COARSE_LOCATION, per definire la posizione del dispositivo
- INTERNET, per connettersi con i servizi Firebase

Patrick Nicolosi X81000928