

Smart Lock – Human Computer Interaction



Edoardo Cagnes, 6435253
Andrea Giorgi, 6430958

Abstract

Lavorando presso la nostra azienda, abbiamo notato che occasionalmente i colleghi si trovano ad avviarsi verso il bagno per poi trovarlo occupato. Ci siamo quindi chiesti quando ed in quali altre situazioni sarebbe comodo sapere se la serratura di una porta sia chiusa o aperta. Con queste domande in mente, abbiamo avviato lo studio per la realizzazione di uno strumento in grado di monitorare lo stato di una serratura.

1. Introduzione

In questa sezione, oltre alla descrizione dell'idea che ha portato alla realizzazione di Smart Lock, nome che abbiamo scelto per sistema da noi progettato, vengono mostrate alcune delle tecnologie usate. Successivamente tali scelte saranno giustificate all'interno delle sezioni di progettazione.

1.1 L'idea

Nell'era dell'Internet of Things, ci siamo chiesti come contribuire a questa realtà attraverso un'idea semplice ma allo stesso tempo efficace.

Le serrature, ormai onnipresenti attorno a noi, stanno anch'esse subendo un processo evolutivo grazie all'influenza di questo mondo e, semplicemente osservando l'ambiente lavorativo, ci siamo resi conto dell'utilità del poterne conoscere lo stato.

Questo concetto non è altro che il punto di partenza verso una visione più ampia che può includere un gran numero di dispositivi intelligenti monitorabili a distanza volti a migliorare le azioni di tutti i giorni.

1.2 Android

Android è il sistema operativo per dispositivi mobili più diffuso al mondo [1], è basato sul kernel Linux ed è sviluppato da Google LLC.

Le applicazioni native per questo sistema sono sviluppate principalmente in linguaggio Java (recentemente anche in Kotlin) e sono organizzate in viste denominate "Activity" strutturate in base a dei "Layout" in formato xml.

Ogni Activity può comporsi di più strumenti atti all'interazione con l'utente. Possono essere pulsanti, liste, immagini o qualsiasi altro tipo di oggetto interattivo, da cose molto semplici a strumenti complessi.

1.3 ESP8266

Per approcciarsi ai progetti nell'ambito IoT, il chip ESP8266 è uno dei migliori strumenti disponibili: ha un modulo WiFi 2.4 Ghz integrato, il supporto completo al protocollo TCP/IP [2] ed un costo ridotto (inferiore ai 10€). Sono disponibili sul mercato diverse versioni (alcune anche estremamente compatte), ma per la realizzazione di questo progetto è stata utilizzata la variante NodeMCU, contenente il modulo ESP-12e

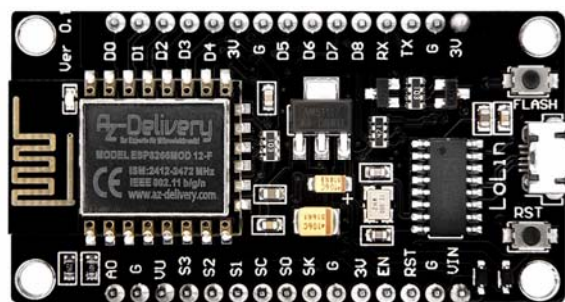


Fig 1 – NodeMCU con ESP-12e

Dotato di 9 input pin digitali, un analogico ed un led incorporato, ha una memoria flash di 4MB: più che sufficiente per il nostro progetto. Inoltre, può essere alimentato e programmato mediante un semplice cavo USB. Solitamente utilizzato in combinazione con altre schede programmabili (Arduino, Raspberry), anche da solo offre tutto il necessario per una grande varietà di progetti.

2. Needfinding

Una volta individuata l'idea, è stato fatto un processo di NeedFinding: si è passati prima per la costruzione di domande accurate, poi alla creazione di Persona ed infine all'individuazione di scenari e requisiti.

2.1 Interviste

Una parte importante degli intervistati sono gli impiegati dell'azienda di sviluppo software "Firenze Web Division S.r.l.". I restanti intervistati sono i titolari del negozio di ferramenta "Mesticheria Aglianese", al fine di ottenere il punto di vista anche di potenziali venditori del prodotto da noi sviluppato.

In totale sono state intervistate 14 persone.

2.2 Persona

Dalle interviste effettuate sono state costruite 3 Persona

- **Impiegata d'ufficio Serena (30-45 anni)**

Serena è un'impiegata d'ufficio in una compagnia che produce software ed applicativi web. Essendo molto occupata e dovendo prendersi cura dei figli, spesso dimentica se abbia eseguito o meno azioni di tutti i giorni come, ad esempio, chiudere la porta di casa. In particolare, spiega che conoscere lo stato di qualcosa può essere molto utile per minimizzare i tempi di attesa in diversi ambiti: per esempio, conoscendo il numero di postazioni occupate dal parrucchiere, potrebbe stimare quanto tempo dovrà attendere prima del suo turno. Infine, nota che una simile funzionalità potrebbe rivelarsi molto utile per controllare lo stato anche di altre cose (gas, frigorifero ...) principalmente in ambito safety.

Usa unicamente dispositivi Android.

- **Studente lavoratore Umberto (19-25 anni)**

Umberto è uno studente che sta facendo esperienza professionale presso un'azienda produttrice di software e applicativi web. Dal suo punto di vista, conoscere lo stato delle cose per ottimizzare i tempi di attesa è un concetto molto interessante, poiché per i molteplici impegni si trova spesso costretto ad incastrare le sue attività. Sostiene che questo tipo di sviluppo possa essere il passo iniziale per arrivare a controllare ed alterare lo stato delle cose direttamente tramite l'uso di dispositivi intelligenti (per esempio aprire o chiudere una porta con un cellulare). Sembra molto più interessato al concetto di

ottimizzazione dei tempi piuttosto che a quello della safety.

Utilizza sia dispositivi iOS che dispositivi Android.

- **Titolare di negozio di ferramenta Fabio (25-35 anni)**

Fabio è il titolare di una mesticheria e vende la maggior parte delle serrature di uso comune. Dal suo punto di vista, l'utilizzo primario di un dispositivo per il controllo dello stato di una serratura dovrebbe essere in ambito di safety: a volte capita anche a lui di dimenticarsi la porta di casa aperta. Ci racconta inoltre che, non molto tempo prima della nostra intervista, ha dovuto aspettare che qualcuno tornasse ad aprire la porta di casa per lui poiché aveva dimenticato di prender le chiavi. Se lo avesse saputo a priori si sarebbe organizzato diversamente, magari svolgendo impegni invece che attendere davanti ad una porta. Essendo un venditore di serrature, gli abbiamo fatto qualche domanda più specifica: ci comunica che, dal suo punto di vista, un prodotto simile potrebbe essere molto valido: sarebbe disponibile ad investire su di esso e ad acquistarne alcune unità per venderle ai suoi clienti.

Comparandolo con buone serrature di marca, stima che un prodotto del genere potrebbe essere venduto ad un prezzo di 45/50€.

Usa solo terminali Android.

2.3 Scenari e Requisiti

Per ogni Persona individuata nel paragrafo precedente, è stato costruito uno scenario al fine di ricavare i requisiti che l'applicazione dovrà soddisfare:

- **Scenario 1 - la giornata di Serena**

Serena si alza presto la mattina per svolgere diverse attività prima di andare in ufficio. Uscita di casa, normalmente chiude la porta a chiave dietro di sé ma, essendo molto indaffarata, spesso si dimentica di farlo. Ed è quello che succede proprio oggi: dopo aver lasciato il figlio a scuola, il dubbio di aver chiuso la porta o meno inizia a farsi strada tra i suoi pensieri. Serena in effetti è un po' paranoica, e un recente furto in un'abitazione nella sua zona non aiuta certamente a tranquillizzarla. Purtroppo però **non ha la possibilità di tornare a casa per verificarlo di persona**: oggi ha un'importante scadenza ed il non è tollerato neanche il minimo ritardo. Fortunatamente però, ha da tempo cominciato ad

utilizzare SmartLock e, con un semplice tap, sul suo smartphone può verificare di averla chiusa correttamente.

• Scenario 2 – gli impegni di Umberto

Umberto tutte le mattine prende il treno per andare in ufficio. Oggi arriva abbastanza presto e decide di controllare con il suo smartphone se i colleghi siano già arrivati: anche se in anticipo, se l'ufficio fosse già aperto preferirebbe mettersi subito a lavoro per avvantaggiarsi, visto che il pomeriggio dovrà passarlo in Università. Poiché sulla porta principale è installato Smart Lock, con un semplice tap scopre che nessuno ha ancora aperto e si concede una piccola pausa al bar per fare colazione. Controlla nuovamente dopo poco e nota che, questa volta, la serratura dell'ufficio risulta aperta. Comincia quindi la sua mattinata lavorativa, saluta i colleghi e si siede alla sua postazione.

Dopo un paio d'ore si concede una pausa e decide di fare una sosta al bagno. Prima di andare però, per evitare inutili attese decide di controllare con Smart Lock per verificare se sia libero o occupato.

• Scenario 3 – Fabio, un acuto venditore

Fabio apre la sua mesticheria verso le 8 del mattino. Tra i molti prodotti che vende, quelli sicuramente più interessanti sono i dispositivi IoT: piccoli device che si integrano nella quotidianità. Per poterli proporre ai suoi clienti, tali dispositivi devono essere facili da installare e intuitivi da utilizzare. La novità di oggi è Smart Lock, un piccolo accessorio per le serrature che consente di monitorarne lo stato a distanza. Pochi minuti dopo arriva il primo cliente, e Fabio gli propone la novità. Il cliente è un po' dubbioso: è convinto di dover smontare parte della sua porta per inserirlo.

Fabio lo rassicura: SmartLock si installa in maniera semplice nel buco già esistente del chiavistello, si connette alla rete di casa e comunica con lo smartphone. In merito a quest'ultimo punto, l'associazione Smart Lock/smartphone avviene attraverso una rapida "scansione".

Il cliente è soddisfatto e decide di provare Smart Lock.

Negli scenari sopra descritti sono stati evidenziati gli aspetti chiave (in carattere sottolineato) dai quali sono stati estratti i seguenti requisiti:

- il sistema Smart Lock deve poter essere utilizzato a distanza;

- il controllo dello stato della serratura deve essere semplice ed immediato;

- lo stato della serratura deve essere controllabile dallo smartphone;

- il dispositivo hardware deve essere piccolo e facile da installare;

- si deve collegare in maniera semplice all'applicazione.

Sulla base dei sopra elencati requisiti e sulla base delle Persona illustrate, è stato deciso di realizzare l'applicazione per dispositivi Android (molto più diffusi ed usati rispetto ai sistemi iOS); per questo motivo, oltre a questi requisiti specifici del prodotto Smart Lock, durante la realizzazione sono stati comunque tenuti presenti i requisiti base di ogni applicazione Android: responsività, intuitività e look accattivante.

3. Progettazione

Concluso un accurato Needfinding, Smart Lock entra nel processo di progettazione. Lo strumento a nostro avviso migliore per la realizzazione di progetti più o meno accurati è il buon vecchio blocco note cartaceo. Per un progetto non eccessivamente complesso come questo, riteniamo che non ci sia bisogno di utilizzare strumenti eccessivamente articolati.

Tutta la fase di progettazione è stata svolta tenendo a mente un concetto principale: il sistema Smart Lock (App e Device) deve essere estremamente semplice ed intuitivo.

3.1 Wireframe

In prima istanza, sono stati realizzati degli schizzi approssimativi del dispositivo hardware da interfacciare con la serratura:

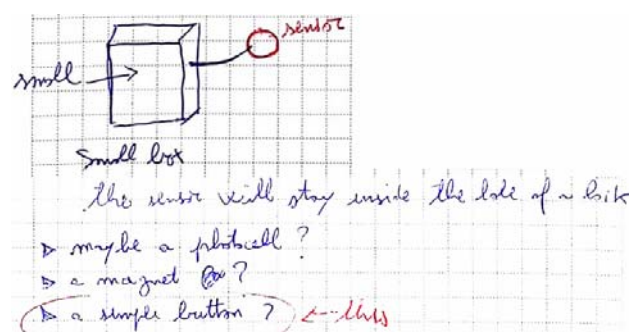


Fig 2 – Primi schizzi del dispositivo

Il concetto è semplice: il dispositivo hardware processa le informazioni fornite da un sensore inserito nella porta per capire se la serratura sia chiusa o meno.

Inizialmente l'idea era di utilizzare una fotocellula inserita nel buco del chiavistello, ma in questo caso sono stati individuati due principali problemi:

1- in caso di assenza di luce non è possibile rilevarne lo stato (si pensi ad esempio alla porta d'ingresso principale di una casa, la notte)

2- più che della serratura, avrebbe rilevato lo stato di apertura della porta (nel buco del chiavistello non c'è luce a prescindere dallo stato della serratura, il che rende impossibile rilevarne lo stato)

L'idea finale è stata quella di inserire un piccolo pulsante tattile nel buco del chiavistello: quando l'utente gira la chiave, il chiavistello preme sul pulsante mandando un segnale al dispositivo hardware.

Una volta decise le basi del dispositivo hardware, è stato realizzato un Wireframe dell'applicazione Android.

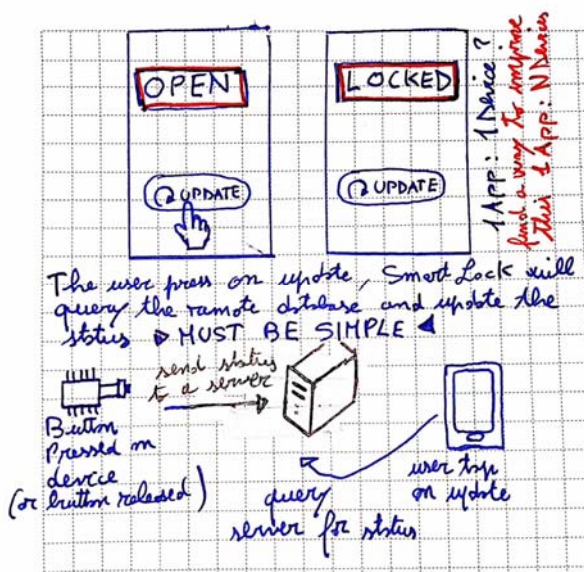


Fig 3 - Smart Lock Wireframe

L'applicazione deve effettuare poche azioni, ma deve farlo bene. Per questo motivo, è importante avere una vista principale dedicata interamente alla visualizzazione dello stato della serratura: l'utente preme su un pulsante e lo stato viene automaticamente aggiornato.

Immaginando Smart Lock come un prodotto commerciale, durante questa fase ci siamo resi conto di quanto un unico dispositivo hardware possa essere limitante: nelle fasi successive sarà necessario tenere in considerazione la possibilità di aggiungere all'applicazione più dispositivi.

3.2 Mockup

Una volta definite le basi, si può passare ad un'analisi più dettagliata del sistema Smart Lock attraverso dei mockup.

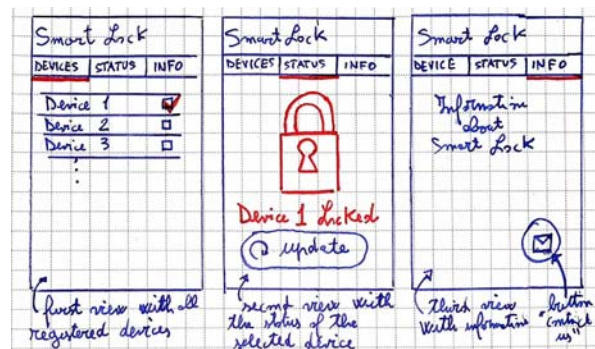


Fig 4 - App Android Mockup

L'applicazione sarà formata da 3 viste intercambiabili attraverso uno slider orizzontale posto in alto.

Nella prima vista l'utente potrà selezionare il dispositivo di cui vuole controllare lo stato; nella seconda sarà mostrato lo stato effettivo del dispositivo ed un pulsante per aggiornarlo; nella terza, le informazioni su Smart Lock ed un pulsante per i contatti gli sviluppatori.

Il dispositivo hardware, invece, deve essere in grado di connettersi alla rete Internet in modo da poter comunicare lo stato della serratura anche da remoto. Quindi, come già anticipato nella sezione introduttiva, il dispositivo sarà basato sul chip ESP8266. Nello specifico ecco un mockup dettagliato che ne illustra il funzionamento:

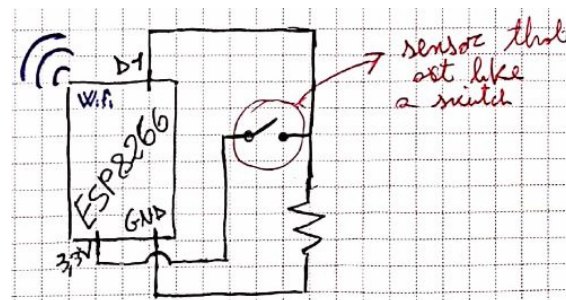


Fig 5 - Device Mockup

Quando il sensore viene attivato, il dispositivo invia un segnale via Internet che indica lo stato della serratura.

Applicazione e dispositivo si interfacciano mediante un server che, attraverso i segnali ricevuti dal ESP8266, è in grado stabilire lo stato della serratura. Per aggiungere invece un dispositivo alla lista dell'applicazione sono state analizzate due ipotesi:

- QR code contenente un codice identificativo del dispositivo;
- Chip NFC contenente un codice identificativo del dispositivo.

Vista la capillare diffusione dei mezzi di pagamento tramite NFC, riteniamo che gran parte degli utenti conosca questa tecnologia e la nostra scelta è caduta quindi su quest'ultima alternativa. Inoltre, tramite la nostra esperienza lavorativa, abbiamo potuto sperimentare diverse tipologie di chip NFC che si sono rivelati estremamente duraturi e resistenti anche a condizioni e situazioni estreme (alcuni anche oltre i 100°C). Per finire, a seconda di dove viene posizionato il dispositivo hardware può risultare molto scomodo inquadrare il QR Code (è sicuramente più pratico avvicinare delicatamente il telefono per la scansione).

Nella prossima sezione sarà illustrato nel dettaglio il primo prototipo del sistema Smart Lock.

4. Dispositivo e Applicazione

4.1 Dispositivo Hardware

Per la realizzazione del primo prototipo è stata utilizzata una breadboard, una griglia che consente di disporre e collegare componenti elettrici senza l'utilizzo di saldature.

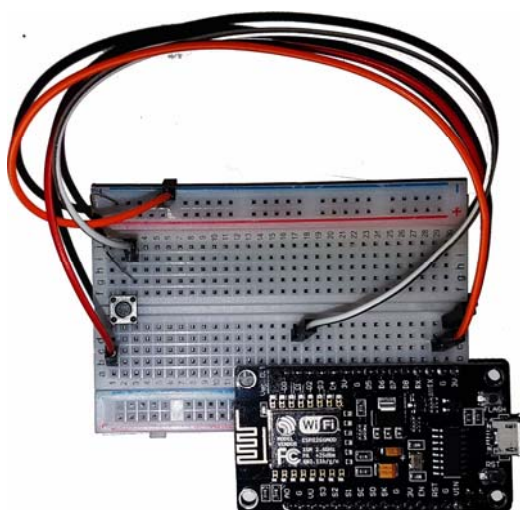


Fig. 6 – Prototipo per test

Sul pin D1 di ESP8266 è collegato un pulsante tattile e, ogni volta che questo viene premuto o rilasciato, un segnale che aggiorna lo stato della serratura viene inviato ad un server remoto.

Attraverso Arduino IDE, il dispositivo ESP8266 viene programmato con il codice da eseguire (in

linguaggio C++). Questo è strutturato in due funzioni principali:

- `setup()` è la funzione che viene eseguita all'avvio del dispositivo. Al suo interno è stato inserito il codice per effettuare la connessione WiFi. Il led di ESP8266 lampeggia ad intervalli di 300 ms fino a connessione avvenuta;
- `loop()` viene eseguita ripetutamente una volta conclusa l'esecuzione di `setup()`. Il dispositivo rileva la pressione del pulsante e provvede ad inviare una richiesta HTTP POST al server remoto contenente un codice identificativo del dispositivo e lo stato della serratura (ovvero se il pulsante è premuto oppure no). Se in qualsiasi momento dovesse cadere la connessione WiFi, il dispositivo provvederà ad eseguire nuovamente il codice per la connessione. Se lo stato non viene modificato per un determinato lasso di tempo (al momento impostato su 2 minuti) viene comunque inviato un segnale al server per indicare che il dispositivo è operativo e connesso alla rete.

Dopo una prima versione utilizzata per testarne le funzionalità, ESP8266 è stato rimosso dalla breadboard ed inserito in un supporto più compatto, all'interno di un contenitore in plastica. I cavi che collegano il dispositivo al pulsante sono stati prolungati in modo da consentirne un corretto posizionamento all'interno del buco del chiavistello. Sul contenitore è stato posizionato un chip NFC contenente il codice identificativo del dispositivo in modo da poterlo associare con l'applicazione. Sopra di questo, al fine di farlo risaltare, è stato applicato un adesivo identificativo.



Fig. 7 – adesivo NFC

4.2 Server e Database

Il server espone due script PHP che consentono l'interazione tra l'applicazione ed il dispositivo:

- `updateStatus.php` è il file che riceve i messaggi inviati dal dispositivo hardware e salva lo stato all'interno di un database SQL. I dati salvati sono: lo stato, il codice identificativo del dispositivo e l'ora in cui il segnale è stato ricevuto;

- `monitor.php` viene chiamato dall'applicazione Android e utilizzando le informazioni contenute nel database restituisce lo stato corretto della serratura. Utilizza anche il campo contenente l'ora dell'ultimo segnale per stabilire se il dispositivo è online o no.

Le connessioni al database sono tutte eseguite con PHP Data Objects (PDO) [3] al fine di garantire una maggiore usabilità e sicurezza.

4.3 Applicazione Android

L'app è sviluppata interamente in linguaggio Java mediante l'utilizzo dell'IDE Android Studio [4]. È strutturata in una activity principale contenente 3 viste denominate "fragment".

Per la realizzazione della grafica è stato scelto un tema scuro, ottimale anche dal punto di vista dei consumi poiché con i moderni display a led, i colori scuri corrispondono ad un utilizzo inferiore della batteria [5].

All'avvio, l'applicazione mostra la vista dei dispositivi disponibili. Al primo avvio questa chiaramente è vuota e richiede di associare almeno un dispositivo mediante scansione NFC: sarà sufficiente avvicinare lo smartphone al chip posizionato sul dispositivo hardware per aggiungerlo alla lista. Una volta effettuata l'associazione, questo comparirà nell'oggetto `ListView` del fragment:

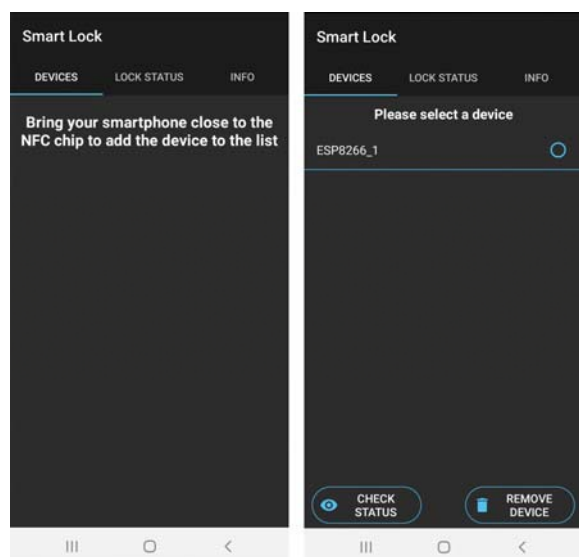


Fig. 8 – Vista dei dispositivi prima e dopo la scansione NFC

A questo punto l'utente potrà rimuoverlo dalla lista dopo averlo selezionato premendo sul pulsante "Remove Device", oppure potrà controllarne lo

stato premendo su "Check Status". In quest'ultimo caso, l'activity si sposterà sul fragment per la visualizzazione dello stato.

Questo è formato da un'immagine figurante un led simile ad un semaforo che indica lo stato del dispositivo. Sotto di questo vi è una breve descrizione dello stato.

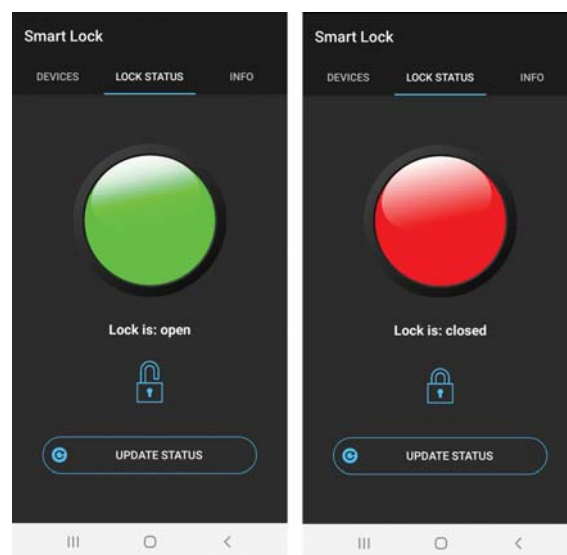


Fig. 9 – Vista dello stato con serratura aperta e chiusa

Il pulsante "Update Status" consente di aggiornare lo stato della serratura.

Nell'ultimo fragment a destra, oltre ad avervi inserito le informazioni su Smart Lock, è presente anche un pulsante per inviare una mail direttamente agli sviluppatori.

L'utente può muoversi tra i fragment effettuando swipe a destra o a sinistra, oppure premendo sulla barra di navigazione in alto.

Al fine di ottimizzare l'esperienza utente, tutte le connessioni di rete sono effettuate attraverso task asincroni estendendo la classe `AsyncTask` di Android. In questo modo, l'utente può continuare ad utilizzare l'applicazione senza dover aspettare la risposta delle connessioni.

L'applicazione supporta il monitoraggio di più dispositivi hardware: nella versione finale, sarà possibile controllare più di una serratura semplicemente effettuando la scansione NFC di più dispositivi.

L'applicazione guida anche l'utente attraverso dei meccanismi correttivi in caso di azione non previste:

- se l'utente preme sui pulsanti "Check Status" o "Remove Device" senza aver selezionato un dispositivo dalla `ListView`, l'applicazione visualizzerà un messaggio Toast;

- se l'utente preme sul pulsante "Update Status" dal fragment per la visualizzazione dello stato senza però aver selezionato un dispositivo, l'applicazione riporterà l'utente sulla vista contenente la ListView dei dispositivi e, come nel caso precedente, visualizzerà un messaggio Toast;
- se il dispositivo è offline, il fragment per la visualizzazione dello stato mostra l'immagine figurante il led di colore grigio e la dicitura "Device is offline";
- se lo smartphone è offline e l'utente prova ad eseguire qualsiasi operazione di controllo dello stato, l'applicazione visualizzerà un apposito messaggio Toast.

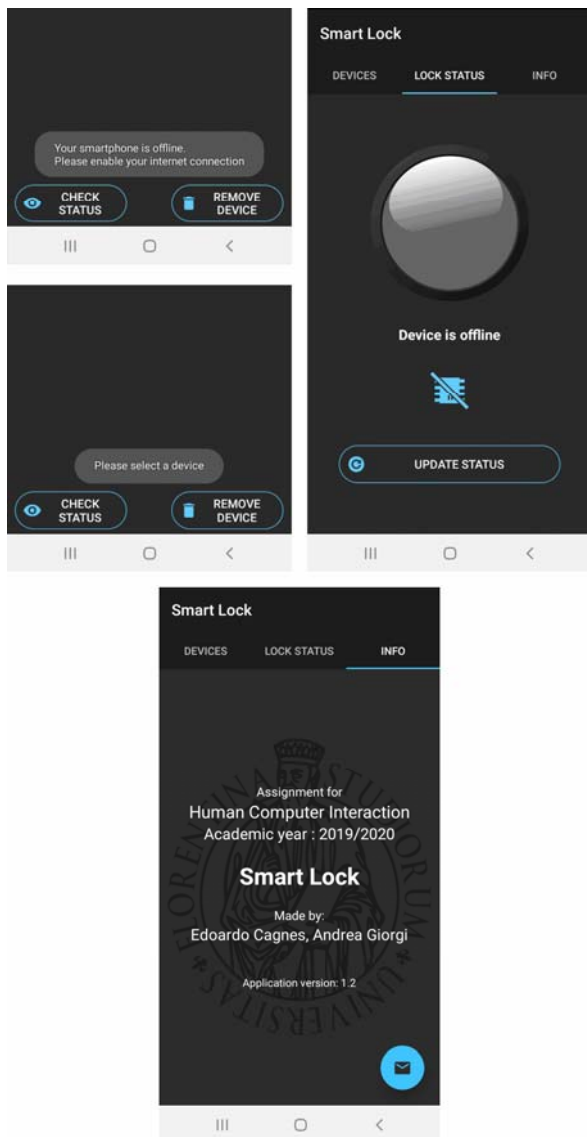


Fig. 10 – Messaggi Toast, stato del dispositivo offline e fragment delle informazioni

5. Test di usabilità

Conclusa la realizzazione del primo prototipo, la fase di test si è svolta presso gli uffici della "Firenze Web Division S.r.l.", facendo utilizzare il sistema Smart Lock da parte degli utenti selezionati per le interviste in fase di needfinding.

Dopo averne discusso, la scelta è ricaduta sull'esecuzione di "Scripted Moderated Test": tenendo conto di quanto indicato in [6], nel nostro caso non allontaniamo gli utenti dalla comfort zone, in quanto i test si svolgono nell'ambiente lavorativo di tutti i giorni e questo è, a nostro avviso, un importante vantaggio. Inoltre i test possono essere svolti semplicemente in orario lavorativo, sfruttando le regolari pause senza richiedere quindi tempo o costi aggiuntivi. Allo stesso modo la disponibilità degli utenti non è un problema: tutti gli impiegati possono contribuire ai test.

Sono stati definiti i seguenti goal:

- l'utente ha appena comprato Smart Lock e vuole installare il dispositivo hardware;
- una volta installato il dispositivo l'utente deve inserirlo nell'applicazione;
- l'utente controlla lo stato della serratura;
- l'utente utilizza la serratura con il dispositivo installato.

Una volta stabiliti gli obiettivi, il test si è svolto nel seguente modo:

- ad ogni utente è stato consegnato un breve foglio di istruzioni in merito all'installazione iniziale del dispositivo (quello che ipoteticamente potranno trovare nella confezione in cui è venduto Smart Lock)

Guida per l'installazione

- 1) inserisci il pulsante dentro il buco del chiavistello
- 2) alimenta il dispositivo con un cavo usb
- 3) dopo aver installato SmartLock scansiona il chip NFC con il tuo smartphone

- nella fase iniziale è stato simulato il montaggio del dispositivo. Ad ogni utente è stato fatto installare il pulsante all'interno del foro del chiavistello sulla porta del bagno;

- In seguito, dopo aver installato Smart Lock sui dispositivi Android utilizzati per lo sviluppo software, ad ogni utente è stato chiesto di aprire

l'app Smart Lock e di procedere con la scansione NFC come indicato sia all'interno dell'applicazione che nelle istruzioni da noi fornite;

- A turno, è stato loro chiesto di entrare nel bagno e di chiudere la serratura, attivando quindi il dispositivo. Chi rimaneva all'esterno poteva controllare con l'applicazione quando la porta sarebbe stata riaperta.

Se per il dispositivo sono state fornite istruzioni, questo non è avvenuto per l'applicazione Android, al fine di verificarne l'intuitività: l'utente deve essere in grado di apprendere da solo il funzionamento di quest'ultima.

Per i test sono stati selezionati 6 utenti: secondo quanto specificato in [6], coinvolgere più di 5 utenti sarebbe uno spreco di risorse ma, poiché nel nostro caso aggiungere un utente non avrebbe comportato alcun costo o sforzo aggiuntivo, abbiamo preferito includerlo nella sperimentazione.

Durante i test, oltre alle impressioni generali degli utenti, sono stati raccolti dei parametri quantitativi in merito all'installazione del dispositivo e all'utilizzo dell'applicazione:

| Parametro | Valore |
|--|------------|
| Tempo per l'installazione | 90 ~ 120 s |
| Tempo per l'associazione | < 10 s |
| # di click su "Check Status" senza aver selezionato il dispositivo | 2 su 6 |

Tab. 1 – Test quantitativi dell'applicazione

L'installazione del dispositivo è abbastanza rapida e, a parte il caso di un utente che per errore ha inserito il pulsante nel buco della scrocca anziché del chiavistello, si è svolta sempre senza problematiche. Al fine di evitare il ripetersi questo errore, sarà sufficiente modificare il punto 1 della "Guida per l'installazione" sopra citata come segue:

1) inserisci il pulsante dentro il buco del chiavistello e NON nel buco dello scrocco

Eventualmente inserendo anche un'immagine esplicativa.

Il tempo per l'associazione del dispositivo è pressoché immediato: come previsto, con l'avvento dei metodi di pagamento NFC come Google Pay, ormai molti utenti hanno confidenza con questa tecnologia ed hanno compreso subito come effettuare la scansione correttamente.

Il merito all'ultimo test citato in Tab. 1, sebbene due utenti abbiano provato a premere sul pulsante "Check Status" senza aver selezionato il dispositivo, i meccanismi correttivi implementati, descritti nella sezione 4.3, li hanno guidati a compiere le giuste operazioni.

Conclusa la fase di test, ad ogni utente è stato consegnato un questionario di tipo SEQ con domande tecniche e non, relative alle azioni compiute e all'esperienza d'uso. Agli utenti è stato richiesto di rispondere ad ogni domanda con un punteggio da 1 (fortemente in disaccordo) a 7 (fortemente d'accordo). I risultati sono mostrati in Tab.2.

Raccolti ed analizzati i questionari, agli utenti è stato chiesto il perché di alcune delle loro risposte:

- alcuni utenti hanno risposto che il sistema è ingombrante nonostante gli sforzi per tenere il dispositivo il più compatto possibile. Questo perché il prototipo è al momento alimentato da una batteria portatile per smartphone molto capiente (e quindi molto grande). In futuro, questo aspetto sarà sicuramente corretto implementando una batteria compatta integrata;

- un utente, in merito alla domanda n° 10 (Tab. 2), ha risposto 6 in contrasto con il voto espresso da tutti gli altri utenti. Una volta interrogato sul perché, questo ha risposto di aver frainteso la domanda, ritenendo difficile smantellare il sistema Smart Lock dalla serratura e non la rimozione del dispositivo dall'applicazione.

Conclusa anche questa fase, agli utenti è stato chiesto che tipo di miglioramenti ritengono sia opportuno effettuare su questa versione del sistema Smart Lock. Gli aspetti che, a detta degli utenti, richiedono miglioramenti sono due:

- Aggiornamento dello stato automatico. Il sistema deve essere in grado di aggiornarsi in autonomia ed eventualmente notificare all'utilizzatore quando lo stato della serratura cambia da chiuso ad aperto e viceversa;
- l'utilizzatore deve essere in grado di configurare autonomamente i parametri per la connessione WiFi senza doverli cambiare dal codice sorgente di ESP8266.

Il sistema Smart Lock torna quindi in fase di progettazione con lo scopo di implementare queste due funzionalità aggiuntive.

| N° | Questions | Mean | σ |
|----|---|------|----------|
| 1 | Il dispositivo è facile da installare? | 6 | 0.58 |
| 2 | Il dispositivo è ingombrante? | 4 | 1.53 |
| 3 | Dopo l'installazione la serratura ha subito qualche peggioramento? | 2.7 | 1.68 |
| 4 | Il sensore ostruisce in qualche modo il movimento del chiavistello? | 3.2 | 2.11 |
| 5 | È stato semplice aggiungere il dispositivo all'applicazione Android? | 6.7 | 0.75 |
| 6 | Ti piace il design dell'applicazione? | 5.4 | 1.36 |
| 7 | L'applicazione è intuitiva? | 6.8 | 0.37 |
| 8 | È difficile controllare se la serratura è aperta o chiusa? | 1 | 0 |
| 9 | Lo stato della serratura è chiaro? | 6.8 | 0.37 |
| 10 | È difficile rimuovere un dispositivo dall'applicazione? | 2 | 1.82 |
| 11 | Hai imparato in fretta ad usare l'applicazione? | 6.8 | 0.37 |
| 12 | L'applicazione presenta dei bug? | 1 | 0 |
| 13 | L'applicazione presenta dei freeze/rallentamenti? | 1 | 0 |
| 14 | Hai mai incontrato falsi positivi o negativi (es. serratura chiusa nell'app ma aperta nella realtà)? | 1 | 0 |
| 15 | Il sistema nel complesso è semplice da usare? | 6.7 | 0.22 |
| 16 | Consigliaresti Smart Lock ai tuoi amici? | 6.5 | 0.5 |
| 17 | Pensi che un simile dispositivo potrebbe essere utile per monitorare altre situazioni (es gas, automobile, frigorifero, etc.) ? | 7 | 0 |

Tab. 2 - risultati dei questionari SEQ. Il parametro σ è la deviazione standard.

6. Miglioramenti

Per implementare le funzionalità mancanti individuate nella sezione precedente è stato necessario intervenire sia sul dispositivo hardware che sull'applicazione Android.

6.1 Mockup / Prototype

Al fine di consentire l'inserimento di parametri di connessione WiFi, è necessario aggiungere un supporto I/O. La scelta è ricaduta su un lettore di schede micro SD contenente un file di testo "wifi.txt" modificabile dall'utente. La prima riga di tale file corrisponde al SSID, la seconda alla password.

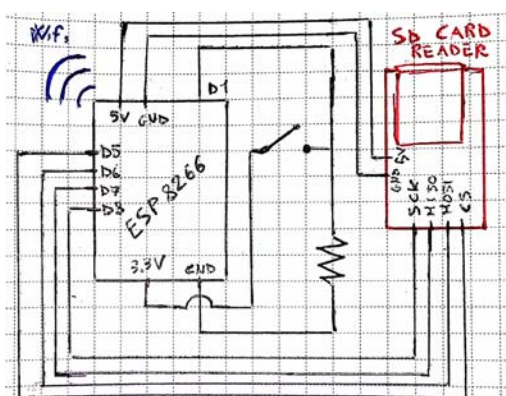


Fig. 11 - Mockup con lettore schede micro SD

La connessione tra ESP8266 e il lettore di schede micro SD avviene attraverso Serial Peripheral Interface [7], un bus standard di comunicazione ideato dalla Motorola. La trasmissione avviene tra un dispositivo detto "master" (ESP8266) e uno o più "slave" (la scheda SD). Il master controlla il bus, emette il segnale di clock e decide quando iniziare e terminare la trasmissione. È una comunicazione seriale, sincrona e full duplex (il master può ricevere e trasmettere contemporaneamente). Le linee utilizzate per la comunicazione sono le seguenti:

- MOSI (Master Out Slave In), la linea utilizzata dal master per inviare i segnali allo slave;
- MISO (Master In Slave Out), la linea utilizzata dallo slave per inviare segnali al master;
- CS (Chip Select) il pin utilizzato dal master per selezionare con quale periferica slave comunicare;
- SCK (Serial Clock) il clock emesso dal master per sincronizzare la trasmissione dei segnali.

Dal punto di vista software, la gestione del bus SPI e la lettura/scrittura su micro SD sono interamente gestite rispettivamente dalle librerie SPI.h e SD.h.

6.2 Applicazione

Al fine di poter aggiornare in tempo reale lo stato, è stata implementata la Firebase Cloud Messaging [8] (FCM), una piattaforma per l'invio di notifiche sviluppata da Google.

In fase di installazione, all'applicazione viene assegnato un particolare token che viene poi utilizzato per inviare la notifica.

Al fine di lasciare all'utente la possibilità di attivare o meno le notifiche, sotto il pulsante "Update Status" nel fragment per la visualizzazione dello stato, è stato inserito uno slider (pulsante a scorrimento) per attivarle/disattivarle. Quando viene premuto viene istanziato un task asincrono che invia al server i seguenti parametri:

- Token dell'applicazione;
- Codice del dispositivo monitorato;
- Parametro per l'attivazione o disattivazione delle notifiche.

Questi dati vengono memorizzati all'interno del database SQL.

Ogni volta che il dispositivo cambia il suo stato, il server di Smart Lock invia una richiesta HTTP POST (attraverso le funzioni cURL PHP [9]) ai server di Firebase con il token del dispositivo ed il contenuto della notifica. Se il token è corretto ed attivo, Firebase risponde con un oggetto JSON [10] avente un parametro "success" impostato ad 1 ed invia la notifica.

All'atto pratico, ad ogni cambio di stato, gli smartphone che hanno sottoscritto le notifiche per quel dispositivo (ovvero che hanno attivato lo slider) ne riceveranno una.

Quando arriva una nuova notifica, questa sovrascrive la precedente (se l'utente non ha provveduto ad aprirla/rimuoverla), in modo da visualizzarne sempre soltanto una. Inoltre, il fragment per la visualizzazione dello stato viene aggiornato con quello notificato, se questo è relativo al dispositivo hardware selezionato.

Quando l'utente rimuove un dispositivo dall'applicazione Smart Lock, automaticamente termina anche la sottoscrizione alla ricezione delle notifiche per quel dispositivo specifico.

Quando invece è l'applicazione Android ad essere disinstallata, il server percepisce automaticamente che il token non è più valido (quando invia la notifica l'oggetto JSON di ritorno ha il parametro "failure" impostato ad 1) e provvede a rimuoverlo dal database.

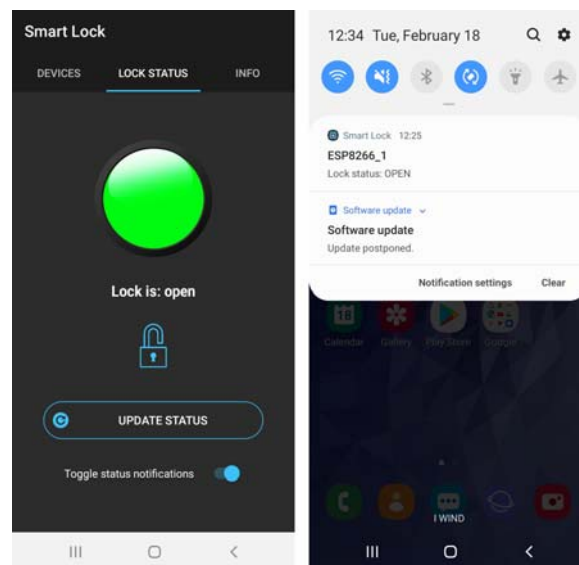


Fig. 12 – Slider e notifica di Smart Lock

7. Test di usabilità parte 2

Anche questi test si sono svolti presso gli uffici della "Firenze Web Division S.r.l.". In questo caso, i test hanno riguardato specificamente le diverse funzionalità implementate.

7.1 Test per la configurazione dei parametri di connessione WiFi

Per questa funzionalità sono stati effettuati "Scripted Moderated Test" come quelli già svolti nella sezione 5.

È stato individuato un goal:

- l'utente connette il dispositivo hardware alla rete WiFi aziendale.

Visto che la funzionalità da testare è una sola, un singolo obiettivo è sufficiente. Il test si è svolto nel seguente modo:

- come nel caso in sezione 5, ad ogni utente è stata consegnata una breve guida per la configurazione WiFi

Guida per la configurazione WiFi

- 1) estrai la scheda micro SD dal vano laterale del dispositivo (sporge fuori dalla scatola in plastica)
- 2) inseriscila nel computer ed apri il file wifi.txt
- 3) nella prima riga inserisci il nome della rete, nella seconda la password (chiave WPA), poi salva le modifiche
- 4) inserisci nuovamente la scheda microSD nel vano laterale del dispositivo

Attenzione agli errori di battitura: qualora il dispositivo non si connetta, controllare di aver scritto tutto correttamente

Se la tua rete WiFi filtra gli accessi mediante indirizzi MAC aggiungi nel tuo router il seguente indirizzo tra i consentiti:

##:##:##:##:##:##

L'indirizzo MAC è oscurato per motivi di sicurezza;

- il dispositivo è stato installato presso la porta del bagno e a turno è stato chiesto ai partecipanti al test di connetterlo alla rete aziendale seguendo le istruzioni fornite ed un adattatore micro SD → USB da noi consegnato.

Una nota in merito ai filtri MAC: si tratta di uno strumento avanzato per la gestione degli accessi alla rete WiFi e siamo convinti che, se un utente è in grado di configurarli, non avrà sicuramente problemi ad aggiungere l'indirizzo di ESP8266 a quelli consentiti.

In questa fase sono stati coinvolti gli stessi utenti selezionati nella sezione 5, ed il test si è svolto senza eccessive problematiche. Si è verificato un singolo caso in cui il dispositivo non si è connesso dopo aver svolto la procedura, ma si è trattato soltanto di un errore di battitura: l'utente si è reso conto dello sbaglio in totale autonomia ed ha corretto i parametri.

Dalle nostre osservazioni sono stati ricavati i seguenti parametri quantitativi:

| Parametro | Valore |
|-----------------------------------|-------------|
| Tempo per la configurazione | 150 ~ 180 s |
| Tentativi di connessioni riusciti | 6 su 6 |

Tab. 3 – Risultati dei test quantitativi.

La procedura è abbastanza veloce: parte del tempo deriva dal fatto che alcuni utenti non ricordavano i parametri di accesso alla rete WiFi ed hanno dovuto recuperarli dalla segretaria.

Conclusi questi test, ai partecipanti è stato chiesto di compilare un breve questionario SEQ di 3 domande (più che sufficienti per testare una sola funzionalità).

| N° | Domande | Media | σ |
|----|--|-------|----------|
| 1 | Configurare i parametri WiFi è semplice? | 5.8 | 0.69 |
| 2 | Un tuo genitore sarebbe in grado di farlo? | 4.5 | 0.96 |
| 3 | Ritieni utile questa funzionalità? | 6.7 | 0.48 |

Tab. 4 - risultati dei questionari SEQ relativi alla funzionalità di connessione WiFi.

I risultati sono mostrati in Tab.4.

Ci aspettavamo un punteggio ancora più basso in risposta al secondo quesito, poiché in Italia c'è la percezione che le persone più anziane abbiano notevoli difficoltà con l'uso della tecnologia. Per il resto, ci riteniamo più che soddisfatti dai risultati ottenuti.

Anche in questo caso abbiamo ascoltato l'opinione dei partecipanti in merito ad eventuali miglioramenti, ed uno di loro ci ha proposto di semplificare ulteriormente l'inserimento dei dati di connessione con l'utilizzo di un piccolo display ed un piccolo tastierino. Tale soluzione è sicuramente interessante e potrà essere valutata per sviluppi futuri.

7.2 Test per la funzionalità di notifica

In merito ai test relativi per il meccanismo di notifica, si è deciso di procedere con test di tipo A/B: i partecipanti sono divisi in gruppi (in genere due) ad ognuno dei quali è consegnata una diversa versione del prodotto da testare [6]. In seguito, il nuovo prototipo è stato installato sulla serratura della porta del bagno degli uffici della "Firenze Web Division S.r.l." e gli impiegati sono stati suddivisi in due gruppi: i primi hanno ricevuto l'applicazione con le notifiche FCM (gruppo A), gli altri senza (gruppo B). In seguito il dispositivo è stato lasciato in funzione per il resto della giornata e i comportamenti dei partecipanti ai test sono stati monitorati.

Conclusa quest'ultima fase di test, dalle osservazioni sono state effettuate le seguenti valutazioni:

- in prima istanza è stata osservata la tendenza da parte degli individui del gruppo A di recarsi al bagno non appena ricevuta la notifica di aggiornamento dello stato (riducendo a 0 i tempi di attesa); è invece capitato qualche volta che gli utenti del gruppo B invece abbiano verificato lo stato sull'applicazione trovandolo occupato (dovendo quindi ricontrollare in seguito). Da questo si evince quindi la maggiore efficienza della soluzione con notifiche FCM rispetto a quella senza;

- **stato vs storico.** Sebbene dal punto di vista della praticità tutti abbiano elogiato il mostrare solamente un'unica notifica contenente l'ultimo stato, alcuni hanno osservato che in certi casi può essere molto utile visualizzarle tutte. Ad esempio nel caso di una stanza il cui accesso deve essere precluso, è importante tenere conto di tutte le

singole variazioni di stato. Una possibile soluzione a questo problema sarà illustrata nella prossima sezione.

Implementati i miglioramenti, come si evince dai risultati delle SEQ e dalle osservazioni a seguito ai test A/B, gli utenti sono generalmente soddisfatti del sistema Smart Lock.

8. Conclusioni e sviluppi futuri

In questo articolo è stato presentato il processo produttivo di Smart Lock, un sistema IoT composto da un dispositivo hardware basato su ESP8266, un web server PHP con database SQL ed un'applicazione Android. Il prodotto svolge correttamente il lavoro per cui è stato progettato e gli utenti risultano esserne soddisfatti.



Fig. 13 - Prototipo finale del dispositivo hardware

Per il rilascio di un'eventuale versione consumer sono stati individuati possibili miglioramenti più o meno complessi:

- memorizzare uno storico delle notifiche ricevute e visualizzarle in una ListView. In questo modo, lasciando la funzione di sovrascrittura delle notifiche nella schermata home, è possibile risolvere la questione stato vs storico;
- chiavistello come conduttore. Utilizzare un pulsante tattile come sensore è un buon compromesso per un prototipo funzionale, ma

nella versione consumer si possono implementare soluzioni ancora migliori. Per esempio, un sottile adesivo in rame da applicare sul chiavistello in modo da fornirgli la capacità di chiudere un circuito quando viene girata la chiave;

- utilizzare cavi piatti flessibili in modo da ridurre qualsiasi tipo di ostruzione che i cavi possono causare al chiavistello;
- cron jobs per notificare lo stato offline. Sempre nell'ambito safety può essere utile notificare immediatamente se un dispositivo va offline (si pensi ad una manomissione). I cron jobs sono task programmati che si eseguono automaticamente ad orari o intervalli prestabiliti.

In conclusione, sono stati individuati possibili sviluppi futuri per il sistema Smart Lock:

- attuatori per chiudere/aprire la serratura. Adesso che è stato ottenuto il pieno controllo dello stato di questa, un possibile passo successivo è essere in grado di chiuderla ed aprirla da remoto;
- controllare lo stato di altri dispositivi. Ci sono moltissimi dispositivi di cui per un motivo o per un altro può far sicuramente comodo conoscere lo stato. Come evidenziato dalla domanda 17 della Tab. 2, tutti gli utenti sottoposti ai test sono d'accordo sull'estendere le funzionalità di controllo anche in altre situazioni.

9. Ringraziamenti

"Firenze Web Division S.r.l." per la collaborazione durante le fasi di Needfinding e Usability Testing.

"Mesticheria Aglianese" per la collaborazione durante la fase di Needfinding.

10. Riferimenti

[1] Market Share Statistics for Internet Technologies
<https://www.netmarketshare.com/operating-system-market-share.aspx>

[2] ESP8266
<https://it.wikipedia.org/wiki/ESP8266>

[3] PHP Data Objects - PHP manual
<https://www.php.net/manual/en/book.pdo.php>

[4] Android Studio
<https://developer.android.com/studio>

[5] AMOLED Black vs. dark gray - XDA Developers
<https://www.xda-developers.com/amoled-black-vs-gray-dark-mode/>

[6] Slide Human Computer Interaction 2019/2020
<https://e-l.unifi.it/course/view.php?id=10891>

[7] Serial Peripheral Interface
<https://www.arduino.cc/en/reference/SPI>

[8] Firebase Cloud Messaging
<https://firebase.google.com/docs/cloud-messaging>

[9] Client URL Library - PHP manual
<https://www.php.net/manual/en/book.curl.php>

[10] JavaScript Object Notation
<https://www.json.org/json-en.html>