



App PuliziaStrade Firenze

Relazione progetto Human Computer Interaction

Pietro Zarri 7042627

Firenze, Maggio 2021



1 Introduzione

L'idea di questo progetto è nata dopo un confronto con alcuni amici sulle problematiche legate al servizio di pulizia strade nel centro di una città molto trafficata come Firenze.

L'obiettivo è stato quello di realizzare un'applicazione per **smartphone** che permetta di conoscere rapidamente i giorni di pulizia strade sul territorio di Firenze e allo stesso tempo fungere da promemoria per avvisare in anticipo gli utenti ed evitare la rimozione del proprio veicolo.

Per realizzare questa applicazione è stato utilizzato il framework **Flutter**, che si basa sul linguaggio di programmazione **Dart**.

2 Needfinding

La prima fase del progetto riguarda il processo di **needfinding**, ossia la ricerca e la scoperta dei requisiti e degli obiettivi da implementare all'interno dell'applicazione.

Sono state dunque fatte delle interviste a possibili utenti, in modo da ricavare informazioni con le quali partire con il lavoro, per poi andare a definire successivamente i vari profili di **personas**.

Le domande che sono state poste agli intervistati sono le seguenti:

- Hai residenza nel Comune di Firenze?
- Quanti veicoli possiedi?
- Per quali scopi utilizzi il tuo veicolo (spostamenti personali, lavorativi, ...) ?
- È facile parcheggiare nella tua zona di residenza?
- Hai difficoltà a rintracciare i cartelli che indicano la data di pulizia strade ?
- Hai mai avuto inconvenienti dovuti al servizio di pulizia delle strade?
- Ricordi quando è la pulizia delle strade dove parcheggi solitamente il tuo veicolo?
- Riesci sempre a ricordare dove hai parcheggiato la macchina?
- Potrebbe essere utile un'applicazione per informare sulle date di pulizia strade e memorizzare i luoghi in cui si è parcheggiato il veicolo?
- Hai praticità nell'utilizzo dello smartphone?

Dalle interviste è emerso che:

- Molte persone che risiedono in centro a Firenze hanno difficoltà a parcheggiare i propri veicoli. Quindi variano spesso la via dove parcheggiano i propri veicoli.
- Ad alcuni intervistati è capitato di subire una rimozione del proprio veicolo dopo avere parcheggiato in una via il giorno di pulizia delle strade.
- Ad alcuni intervistati è risultato difficile rintracciare il cartello che segnala la data di pulizia delle strade.
- Ad alcuni intervistati è capitato di non ricordare la via dove avevano parcheggiato il proprio veicolo.
- Ad alcuni intervistati è capitato di non conoscere il giorno di pulizia della strada nella via in cui avevano parcheggiato il veicolo.
- Molti intervistati sono risultati favorevoli all'utilizzo di un sistema di promemoria per evitare di subire la rimozione del proprio veicolo.
- Ad alcuni intervistati sono risultati interessati ad un sistema per conoscere rapidamente la data di pulizia strade nella via in cui si trovano.
- Ad alcuni intervistati sono risultati interessati ad un sistema di ricerca rapido per conoscere la data di pulizia strade in una determinata via

3 Personas

Dalle interviste è stato possibile delineare tre profili di personas differenti:

- **Studente:** Risiede in centro città e possiede un'auto personale che al ritorno a casa cerca parcheggiare nelle vie circostanti la propria abitazione. Spesso impiega molto tempo a trovare un parcheggio libero.
- **Impiegato:** Possiede due veicoli per gli spostamenti personali e della sua famiglia. Un SUV che parcheggia solitamente nel suo garage privato ed un'utilitaria che parcheggia nei pressi della propria abitazione.
- **Libero professionista:** Risiede fuori città, ma ha un'attività nel centro. Ha un furgone che utilizza alcune volte durante la settimana per la propria attività e che solitamente rimane parcheggiato nei pressi di essa.

4 Scenari

Dopo aver stilato i vari profili di *personas*, sono stati inseriti in degli scenari in modo che, in seguito, ci aiutino a trovare i requisiti adatti alla applicazione:

- **Studente** torna a casa la sera dopo un giorno di lezioni universitarie, parcheggia nel primo posto libero che trova vicino alla sua abitazione.
È molto stanco e non ha voglia di cercare il cartello di pulizia delle strade e spera non sia proprio quella sera.
Avrebbe voluto avere a disposizione uno strumento per conoscere rapidamente il giorno di pulizia in quella via.
- **Studente** il venerdì sera parcheggia la propria macchina in una via e dal cartello capisce che il servizio di pulizia strade è il seguente lunedì.
Torna a casa tranquillo, ma durante il week end organizza alcuni giorni al mare con alcuni amici. Il mercoledì al ritorno scopre che la propria auto è stata rimossa il lunedì scorso.
Avrebbe voluto ricevere un avviso sul telefono per avvisare i suoi genitori e chiedergli di spostare la macchina.
- **Impiegato** sa che ha attualmente parcheggiato la macchina in una via dove è prevista la pulizia strade, deve quindi spostarla in una via dove non è prevista in quella stessa data.
Per evitare di cercare tra molte vie attorno a casa preferirebbe poter conoscere in anticipo la data di pulizia.
- **Libero professionista** lascia sempre il proprio veicolo commerciale parcheggiato nella via antistante la propria attività. Gradirebbe poter controllare con rapidità il giorno in cui è prevista la pulizia in tale strada.

5 Requisiti

I requisiti principali ottenuti tramite l'elaborazione delle interviste e degli scenari, che la web app deve avere sono quindi i seguenti:

- Possibilità di ottenere informazioni sulla posizione in cui ci si trova attualmente tramite **geo localizzazione**.
- Poter **salvare** una determinata via per visualizzare rapidamente le informazioni su di essa.

- Poter **salvare** la posizione dove si è parcheggiato un veicolo.
- Visualizzare le **indicazioni** per raggiungere tale parcheggio.
- Ricevere una **notifica** quando è prevista la pulizia delle strade nel luogo in cui è salvato il parcheggio.
- Consentire una **ricerca** rapida delle strade da un elenco completo di tutte le vie del Comune di Firenze.
- **Visualizzare rapidamente** tutte le strade che sono state aggiunte ai preferiti e poter gestire le notifiche relative a ciascuna di esse.
- Dare all'utente la possibilità di **decidere** con quanto anticipo ricevere una notifica di pulizia strade.

6 Mockup

Dopo aver elaborato i requisiti è stato realizzato un **Mockup** in modo da concretizzare i requisiti ottenuti nello step precedente.

I mockup sono stati realizzati con **Balsamiq Mockups** [1] in versione desktop.

Analizziamo più nel dettaglio le pagine presenti nell'app:

- **Map:** La prima pagina che si presenta all'apertura dell'app è quella della mappa.

La mappa deve presentare solamente gli elementi fondamentali per la localizzare l'utente e individuare il nome delle strade.

È possibile ottenere informazioni sulla pulizia delle strade sulla via in cui ci si trova attualmente tramite un *button*.

In caso una singola via sia suddivisa in più tratti all'interno del dataset, compare un *dialog* che chiede all'utente di selezionare il tratto al quale è interessato.

Le informazioni sono mostrate all'utente tramite un elemento di tipo *bottom sheet*.

Oltre al nome della via e alla data della prossima pulizia strade sono presenti anche due bottoni che permettono di salvare un parcheggio in quella determinata posizione oppure aggiungere la via ai preferiti.

Un altro *button* posizionato nella parte superiore sinistra dello schermo segnala la presenza di un posizione attualmente registrata come parcheggio.

- **Search:** Questa pagina tramite una barra di ricerca consente di cercare una determinata via presente all'interno del database di pulizia strade del Comune di Firenze.

Premendo una via specifica si apre una *card* che mostra la data della prossima pulizia strade e il *button* per aggiungere tale strada ai preferiti.

- **Favourites:** Questa pagina consente all'utente di visualizzare tutte le strade che ha aggiunto ai preferiti, attivare o disattivare le notifiche di pulizia strade relative alla singola via oppure rimuovere una via dai preferiti.

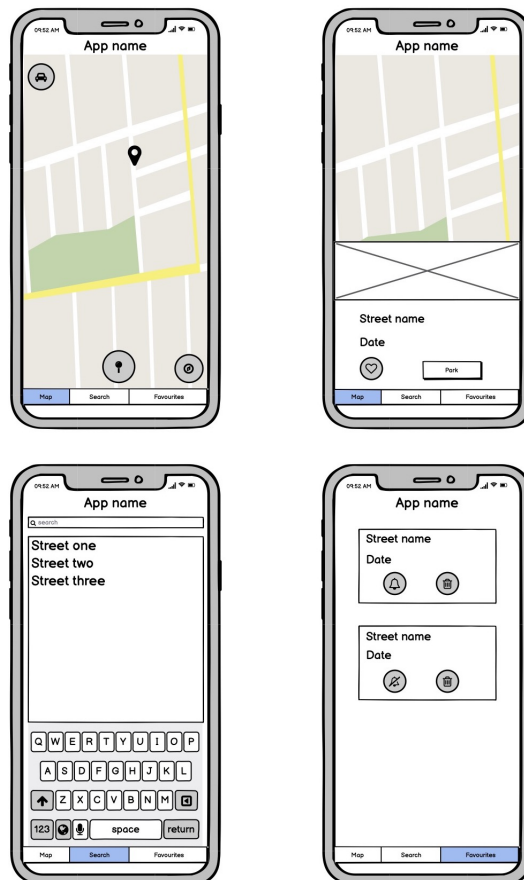


Figure 1: Mockup delle pagine Mappa, Cerca e Preferiti dell'applicazione

7 Implementazione

Dopo avere realizzato il mockup si è passati alla fase di implementazione concreta dall'applicazione.

Il framework che è stato utilizzato per realizzare questa applicazione è **Flutter SDK** [4].

Flutter è un framework open-source creato da Google per la creazione di interfacce native per **iOS** e **Android** e si interfaccia con SDK della piattaforma specifica.

Una particolarità molto apprezzata del Flutter engine, grazie al codice scritto in Dart, è quella di poter effettuare un *hot-reload* dell'applicazione dove la modifica del codice viene iniettata immediatamente all'interno dell'applicazione così da visualizzare all'istante le modifiche effettuate senza un riavvio completo o un cambio di stato.

La progettazione dell'interfaccia utente in Flutter prevede l'assemblaggio e/o la creazione di vari **widget**.

Un widget in Flutter rappresenta una descrizione immutabile dell'interfaccia utente; grafici, testo, forme e animazioni vengono creati utilizzando i widget. È possibile creare widget più complessi **combinando** molti più semplici.

7.1 Dati

I dati sono ottenuti dal sito del **Open Data** del Comune di Firenze [6] che mette a disposizione un file continuamente aggiornato con tutte le vie dove il servizio di pulizia strade è attivo con la relativa data in cui è prevista.

Per accedere ai dati sono state create una serie di API tramite il servizio *Pythonanywhere* [7].

In questo modo l'applicazione può accedere in maniera immediata alle informazioni presenti nel dataset. In particolare sono state realizzate API per:

- **Ottenere elenco** di tutte le vie presenti nel dataset.
- **Conoscere i tratti** in cui è suddivisa una via.
- **Conoscere la data** della prossima pulizia strade in una specifica via e tratto.

7.2 Persistenza

Per mantenere tutte le informazioni necessarie (strade aggiunte ai preferiti, parcheggio salvato, impostazioni dell'app) persistenti sul dispositivo sono stati utilizzati due plugin:

- **sqflite**: è il plugin **SQLite** per Flutter. Supporta sia iOS che Android e consente di eseguire in maniera rapida e immediata operazioni sul database. In particolare sono salvati sul database tutti le strade aggiunte ai preferiti dall'utente. Ciascuna di strada è mappata in un oggetto **PositionInMap**. Ogni **PositionInMap** ha come attributi tre stringhe (nome della strada, nome della sezione, nome della città).

- **shared_preferences**: Flutter plugin per la lettura e scrittura di semplici coppie key-value. È costruito sopra **NSUserDefaults** di iOS e **SharedPreferences** di Android. È utilizzato per salvare le informazioni relative ad un parcheggio memorizzato e le impostazioni utente.

7.3 Mappa

Per integrare una mappa all'interno dell'applicazione sono state utilizzate le **API** di **Google Maps** [5] (Fig. 3a).

Le mappe sono state personalizzate in modo da mostrare solamente le informazioni essenziali necessarie all'utente per l'utilizzo dell'app ovvero il nome delle strade.

Tramite tramite il *package* di Flutter **Location** si riesce ad ottenere con buona posizione la posizione dell'utente e quindi la via in cui si trova.

Sulla mappa sono stati aggiunti sue *floating buttons*:

- Quello posizionato in basso a destra centra la mappa sulla posizione attuale.
- Quello posizionato in basso al centro carica il *bottom sheet* (Fig. 3e) con le informazioni sulla data di pulizia strade e due bottoni per aggiungere/rimuovere una via dai preferiti oppure salvare un parcheggio in quella posizione.

7.4 Parcheggio

Per segnalare all'utente che il parcheggio è stato memorizzato correttamente viene aggiunto un pin alla mappa in corrispondenza della posizione salvata ed inoltre è stato implementato un *expandable floating button* (Fig. 3f).

Se l'utente effettua un tap sul bottone compaiono altri tre bottoni che consentono di effettuare diverse operazioni.

- **Rimuovere** il parcheggio.
- Ricevere **indicazioni** per raggiungere il luogo del parcheggio.
- **Centrare** la mappa nel luogo del parcheggio.

7.5 Cerca

La pagina **cerca** (Fig. 3b) per mettere tramite una barra di ricerca di trovare una specifica via presente all'interno del database di pulizia strade del **Comune di Firenze**.

Una stessa può comparire più volte all'interno dell'elenco se suddivisa in più sezioni all'interno del dataset. Nel sottotitolo, qualora presente, è specificata la sezione.

Per realizzare questa pagina è stato utilizzato principalmente il package **flappy_search_bar**.

7.6 Preferiti

La pagina **preferiti** (Fig. 3c) contiene una **lista** di Card ciascuna delle quali identifica un via che è stata aggiunta ai preferiti dall'utente.

Ogni card presenta **due bottoni**: il primo consente di attivare o disattivare le notifiche per quella specifica via mentre il secondo cancella la via dai preferiti.

7.7 Impostazioni

La pagina di **impostazioni** (Fig. 3d) è stata aggiunta per permette ad un utente di scegliere con quanto tempo di anticipo ricevere una notifica di pulizia strade.

È possibile selezionare tra due, sei, 12 ore o un giorno prima. La pagina di impostazioni è realizzata tramite il package **settings_ui**.

7.8 Notifiche

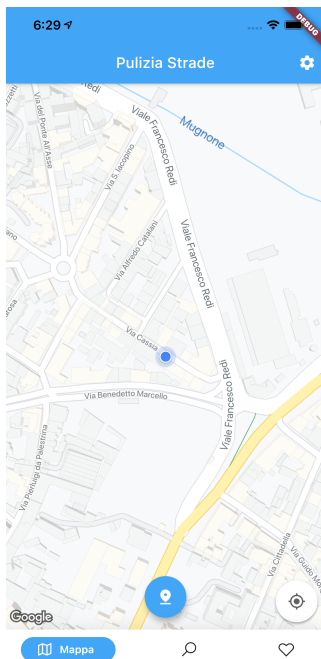
Per implementare un sistema di notifiche all'interno dell'applicazione è stato sfruttato il servizio **Firestore**[3].

In particolare è stata configurata il servizio **Firestore Cloud Messaging** (FCM) [2], che consente di inviare notifiche a dispositivi iOS e Android, programmando il momento di invio della notifica al dispositivo.

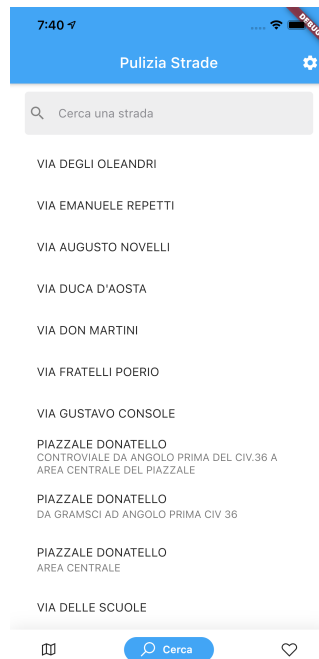


Figure 2: Notifica pulizia strade

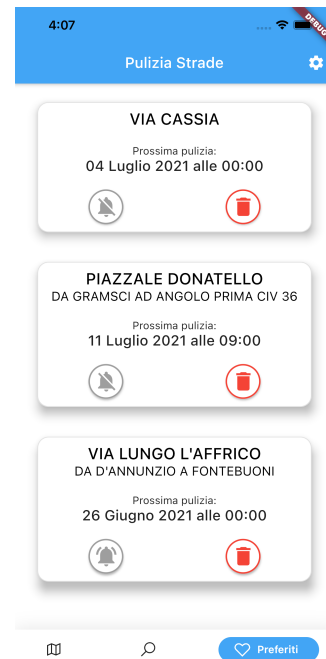
Ogni volta che un utente salva un parcheggio oppure aggiunge una strada ai preferiti viene programmata un notifica per quel dispositivo nel giorno della prossima pulizia strade con un anticipo pari a quello selezionato dall'utente tramite la pagina Impostazioni.



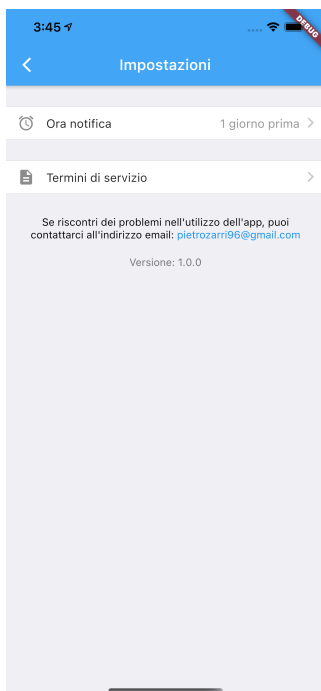
(a) Pagina Mappa



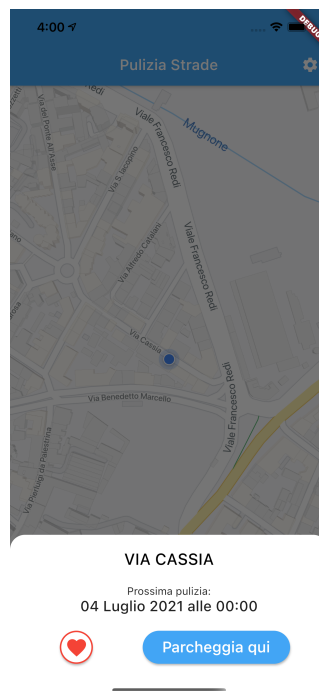
(b) Pagina Cerca



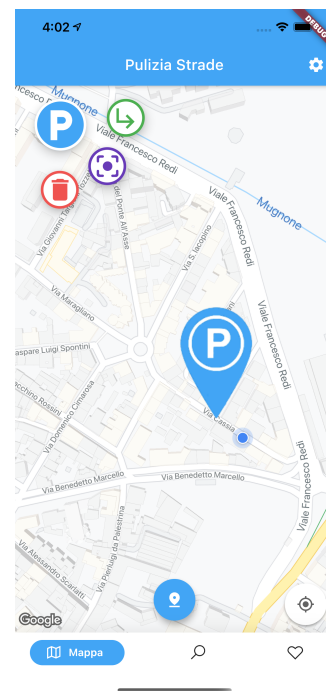
(c) Pagina Preferiti



(d) Pagina Impostazioni



(e) Informazione posizione attuale



(f) Floating button parcheggio

Figure 3: Pagine applicazione

8 Usability testing

Durante la fase finale del progetto sono stati ideati degli usability test, in modo da rilevare potenziali problemi e capire quanto il prototipo costruito risultasse intuitivo e user-friendly ad un utente medio.

8.1 Scelta dei task

Sono stati stabiliti una serie di task per verificare quanto gli utenti sono a loro agio nel flusso di utilizzo dell'applicazione e nell'interazione con il layout e i contenuti.

Dopo qualche minuto iniziale dato per familiarizzare con l'applicazione sono stati fatti eseguire i seguenti task:

1. **Cerca** la data della prossima pulizia strade nella strada in cui ti trovi adesso.
2. **Aggiungi** ai preferiti la strada in cui ti trovi adesso.
3. **Parcheggia** il tuo veicolo nel luogo in cui ti trovi adesso.
4. **Ottieni** le indicazioni tramite "Google Maps" per tornare al luogo dove hai parcheggiato la macchina.
5. **Cancella** il parcheggio.
6. **Cerca** la data della prossima pulizia strade in "Via Cassia".
7. **Aggiungi** Via Cassia ai preferiti.
8. **Disattiva** le notifiche per Via Cassia.
9. **Imposta** le notifiche per la pulizia strade a 12 ore prima.
10. **Rimuovi** Via Cassia dai preferiti.

8.2 SEQ

Per raccogliere informazioni qualitative e quantitative relative allo svolgimento dei task, abbiamo pensato di rivolgere ai candidati anche un questionario, realizzato tramite **Google Moduli**, formato da 22 domande, di cui alcune personali, altre riguardanti i task svolti e altre ancora di carattere generale sull'Applicazione.

Alcune delle domande presentate sono di tipo **SEQ**, con scala di valori che va da 1 a 7, dove 1 indica "**totalmente in disaccordo**" mentre 7 indica "**totalmente d'accordo**".

Altre domande sono a risposta vincolata, e altre ancora a risposta aperta e facultative.

Per evitare che l'utente potesse abituarsi a rispondere sempre con lo stesso punteggio, alcune domande sono state formulate con logica opposta, ad esempio: "ho avuto difficoltà..." (domanda negativa) al posto di "è risultato facile..." (domanda positiva). Le domande del questionario sono le seguenti:

- È stato facile trovare informazioni sulla strada in cui ti trovi adesso?
- È stato facile parcheggiare il veicolo nel luogo in cui ti trovi adesso?
- È stato facile ottenere le indicazioni per tornare al luogo del parcheggio?
- Ho apprezzato poter ottenere indicazioni per tornare al luogo del parcheggio.
- È stato facile interagire con il "expandable button" che segnala la presenza di un luogo salvato come parcheggio?
- È stato facile ricercare una strada tramite la pagina "cerca"?
- È stato facile aggiungere una strada ai preferiti tramite lo specifico bottone?
- Ho apprezzato la possibilità di disattivare le notifiche per una determinata strada.
- È stato facile cambiare il momento di arrivo di una notifica tramite le impostazioni dell'app?
- Quali funzionalità dell'app hai apprezzato particolarmente ?
- Quali aspetti invece hai apprezzato meno ?
- Ho apprezzato la suddivisione delle strade in più zone a seconda del tratto in cui ci trova.

8.3 Pilot testing

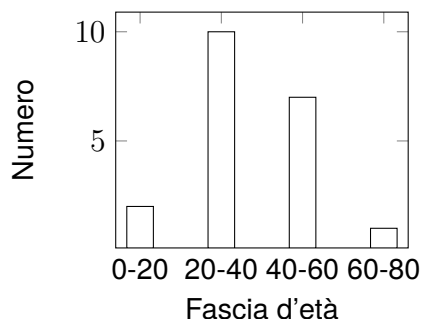
Prima di effettuare i test veri e propri è stato svolto un test pilota, i cui risultati non sono stati analizzati, ma sono semplicemente serviti ad assicurarsi che gli scenari ideati e il questionario a cui sottoporre l'utente fossero facilmente comprensibili e adeguati.

Sono stati riordinate le domande in modo che seguissero il flusso dell'esecuzione dei task. Infatti prima sono eseguiti i test relativi alla pagina **Mappa** e poi quelli relativi alla pagina **Cerca** e **Preferiti**. Inoltre sono state aggiunte alcune domande per rendere il questionario più completo.

- Quale versione dell'app hai utilizzato (iOS/Android)?
- È stato facile trovare informazioni sulla strada in cui ti trovi adesso?
- È stato facile parcheggiare il veicolo nel luogo in cui ti trovi adesso?
- È stato facile ottenere le indicazioni per tornare al luogo del parcheggio?
- Ho apprezzato poter ottenere indicazioni per tornare al luogo del parcheggio.
- Ho avuto difficoltà ad interagire con il "expandable button" che segnala una posizione salvata come parcheggio.
- Ho avuto difficoltà a cambiare pagina tramite il menù inferiore.
- È stato facile ricercare una strada tramite la pagina "cerca"?
- È stato facile aggiungere una strada ai preferiti tramite lo specifico bottone?
- Ho apprezzato la suddivisione delle strade in più zone.
- Ho apprezzato la possibilità di disattivare le notifiche per una determinata strada.
- È stato facile cambiare il momento di arrivo di una notifica tramite le impostazioni dell'app?
- Quali funzionalità dell'app hai apprezzato particolarmente?
- Quali aspetti invece hai apprezzato meno?
- Consigliaresti l'app a persone con esigenze simili?

8.4 Risultati

I test sottoposti sono stati effettuati tutti tramite supervisione e successivamente è stato chiesto di compilare autonomamente il questionario. I partecipanti sono stati 20 con la seguente distribuzione di età:



I risultati delle domande del questionario con risposta con scala di valori sono riportati nella seguente tabella, dove è specificato il valore medio delle risposte e la varianza relativi ad ogni domanda.

Risultati questionario			
	Domanda	Media	σ
1	È stato facile trovare informazioni sulla strada in cui ti trovi adesso?	6.45	0.68
2	È stato facile parcheggiare il veicolo nel luogo in cui ti trovi adesso?	6.7	0.47
3	È stato facile ottenere le indicazioni per tornare al luogo del parcheggio?	6.45	0.67
4	Ho apprezzato poter ottenere indicazioni per tornare al luogo del parcheggio.	6.55	0.68
5	Ho avuto difficoltà ad interagire con "expandable button" che segnala una posizione salvata come parcheggio	3.75	1.3
6	Ho avuto difficoltà a cambiare pagina tramite il menù inferiore	2.5	2.19
7	È stato facile ricercare una strada tramite la pagina "Cerca"?	6.45	0.99
8	È stato facile aggiungere una strada ai preferiti tramite lo specifico <i>button</i> ?	6.65	0.48
9	Ho apprezzato la suddivisione delle strade in più zone	5.25	1.48
10	Ho apprezzato la possibilità di disattivare le notifiche per una determinata via.	6.3	1.03
11	È stato facile cambiare il momento di arrivo di una notifica tramite le impostazioni dell'app?	5.9	0.91

Per quanto riguarda le domande a risposta aperta le **funzionalità più gradite** dell'app sono state:

- Implementazione notifiche.
- Possibilità di salvare un parcheggio e poter indicazioni tramite mappe.
- Ricerca per nome.

Mentre gli aspetti che sono stati apprezzati di meno sono stati:

- Interazione con *button* parcheggio.

- Gestione suddivisione strade.
- Interazione con *button* disattiva notifica.

Per quanto riguarda la domanda “Ci sono delle funzionalità che ti sarebbe piaciuto vedere nell’app”:

- Conoscere data prossima pulizia anche rispetto al luogo dove si è parcheggiato la macchina.
- Aggiunta di altre strade al di fuori del comune di Firenze

9 Miglioramenti

9.1 Button parcheggio

Molti degli utenti durante le prime iterazioni con l’app hanno mostrato delle difficoltà ad interagire con l’*expandable button* che segna un parcheggio memorizzato.

È stato quindi **riprogettato** il componente in maniera che al fianco di ogni *button* sia presente un *label* che in maniera incisiva suggerisca all’utente il comportamento del *button* stesso.

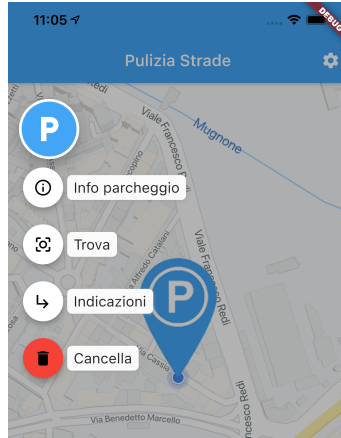


Figure 4: Nuovo button parcheggio

Inoltre adesso il *park button* consente anche di aprire un *dialog* che indica all’utente il nome della via in cui si è parcheggiato e l’eventuale giorno in cui è prevista la pulizia strade.

Per realizzare questo nuovo componente è stato utilizzato il package **flutter_speed_dials**.

9.2 Button notifiche attive

Su suggerimento di alcuni utenti e vista la difficoltà di altri a completare il task numero 8 è stato scelto di cambiare il colore del bottone per disattivare le notifiche su una via. Infatti adesso se le notifiche sono attivate il bottone è di colore verde, quando sono disattivate di colore grigio. Le icone all'interno dei bottoni sono rimaste le stesse.



Figure 5: Button notifiche attivate

9.3 Chiusura dialog

Poiché alcuni utenti avevano difficoltà a chiudere i *dialog* relativi ad una strada tramite un tap al di fuori del del *dialog*. Per permettere di eseguire l'azione in modo più intuitivo è stato aggiunto un *button* "Annulla".



Figure 6: Button annulla

10 Sviluppi futuri e conclusioni

Dopo aver implementato tutte le funzionalità descritte in sezione 9 l'applicazione risulta completa e pienamente utilizzabile sul territorio di Firenze.

Per quanto riguarda altri possibili sviluppi il principale è quello di estendere la funzionalità dell'applicazione ad **altre città italiane**. Ovviamente la difficoltà principale è ottenere dei dati per tutte le altre città che si voglia aggiungere e soprattutto rendere tali dati omogenei fra di loro in maniera da poterli integrare tutti congiuntamente senza variazione nella UX da una città all'altra.

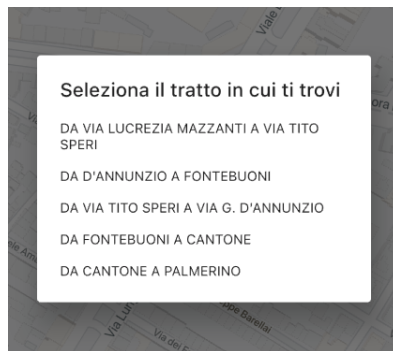


Figure 7: Dialog per scelta sezione

Una modifica interessante per migliorare ulteriormente l'usabilità riguarda quelle strade che sono suddivise in più tratti e dove tratti diversi della stessa via potrebbero avere date di pulizia strade diverse.

Al momento, quando un utente richiede informazioni su una strada di questo tipo si apre un *dialog* che richiede direttamente all'utente di selezionare in quale sezione si trova.

Un modo per eliminare questo step sarebbe quello di provare ad utilizzare un dataset messo a disposizione sul sito **Open Data** del comune di Firenze. Infatti è disponibile un dataset in formato **KML** o **SHP**, che sono applicabili direttamente su un servizio di mappe come Google Maps o OpenStreetMaps.

Incrociando i dati della localizzazione con quelli del dataset applicato alle mappe potrebbe essere possibile risalire alla corretta sezione in cui si trova l'utente senza bisogno del *dialog*.

References

- [1] Balsamiq. <https://balsamiq.com/wireframes/>.
- [2] Firebas cloud messaging. <https://firebase.google.com/docs/cloud-messaging>.
- [3] Firebase. <https://firebase.google.com>.
- [4] Flutter. <https://flutter.dev>.
- [5] Google maps flutter. https://pub.dev/packages/google_maps_flutter.
- [6] Open data comune di firenze. <https://opendata.comune.fi.it/?q=metarepo/datasetinfoid=pulizia-strade>.
- [7] Pythonanywhere. <https://www.pythonanywhere.com>.