

Human Computer Interaction

Street Cleaning Map: sviluppo di una applicazione Android

Edoardo Re
7042226

edoardo.re@stud.unifi.it

Abstract

Nella seguente relazione si descrive lo sviluppo di una applicazione Android che permette di condividere le informazioni sui lavaggi strade tramite una mappa. Gli utenti possono visualizzare le segnalazioni delle pulizie stradali condivise da altri utilizzatori e possono attivare una notifica push che ricordi l'inizio del lavaggio in modo tale da ricordare di spostare la macchina in tempo. Lo sviluppo è stato effettuato su Android Studio ed è stato implementato un data base real time di Google Firebase. Si affronta infine una fase dettagliata di usability testing che ha permesso di evidenziare i punti di forza e le debolezze dell'applicazione

1. Introduzione

La pervasività degli smartphone al giorno d'oggi ha semplificato notevolmente la possibilità di sviluppare dei sistemi interattivi facilmente utilizzabili e che garantiscano all'utente una ottima user experience (UX). Lo scopo principale della Human Computer Interaction (HCI) è quello di studiare e progettare una interazione tra uomo e macchina ottimale che sia semplice ed intuitiva, lo sviluppo di applicazioni Android permette di realizzare un sistema finale con tali caratteristiche per l'utente utilizzatore. In particolare, l'applicazione realizzata permette all'utente di visualizzare dei punti di lavaggio strade già presenti sul Data Base ed inoltre aggiungere nuovi punti e segnalazioni con la possibilità di attivare notifiche push di promemoria. Infine si è realizzata una fase di test per comprendere i punti di forza e le debolezze dell'applicazione sviluppata.

2. Needfinding

Nella prima fase dello sviluppo dell'applicazione si ha il Needfinding ovvero un processo in cui si osservano i candidati per scoprire i loro bisogni. Da questi si costruiscono delle Personas e degli scenari in modo tale da estrarre una prima idea di software.

2.1. Personas

Dalle interviste fatte in fase di needfinding a seguito di un veloce brainstorming sono state create due tipologie di personas:

1. Mario, 26-35 anni, proprietario di un'auto. Si sposta spesso per andare a lavoro e rischia spesso di prendere multe in zone sconosciute a causa del lavaggio strade. Patito di tecnologia, ha uno smartphone che conosce molto bene ed è molto pratico nell'usarlo.
2. Giovanni, 35-50 anni, proprietario di un'auto. Si sposta tutti i giorni in bicicletta, la macchina la usa solamente per andare al mare e la parcheggia spesso in garage privati. Ha uno smartphone ma non è molto pratico nell'utilizzo, è il figlio ad aiutarlo nei problemi quotidiani.

2.2. Requisiti

Le personas create precedentemente sono state collocate all'interno di scenari in modo tale da realizzare ed individuare dei requisiti essenziali da far rispettare all'applicazione da sviluppare. L'applicazione deve risultare intuitiva e rapida, per soddisfare tutti gli utenti come Giovanni che potrebbero non avere troppa dimestichezza con gli smartphone.

3. Mockup

Il Mockup realizzato prima di iniziare a sviluppare l'applicazione ha permesso di chiarire e cercare di riassumere dove posizionare e che aspetto dare all'applicazione una volta terminato il suo sviluppo. Si è deciso di rendere l'interfaccia il più minimale e semplice possibile in modo tale da far arrivare subito l'utente a compiere determinate azioni. La vista principale nel mockup dunque pochi elementi ma ben definiti:

1. Una mappa
2. La posizione attuale GPS

3. Eventuali marker di segnalazione dei lavaggi nelle vicinanze
4. Un pulsante per attivare le notifiche di promemoria
5. Un pulsante per segnalare anche ad altri utenti dei nuovi lavaggi strade

Solamente l'ultimo pulsante permette di accedere ad una nuova view e si è scelto ciò solamente in fase di sviluppo per non andare a sovrapporre nulla alla View principale che risulta omogenea e ben concepita.



Figure 1. Mockup

4. Crowdsourcing

L'applicazione sviluppata ha come fondamento il crowdsourcing, ovvero la collaborazione da parte di più utenti che utilizzano l'applicazione nel condividere le informazioni del lavaggio stradale nella propria posizione. Tramite una apposita view è possibile aggiungere tali informazioni e condividerle con tutti in tempo reale, il corretto inserimento permette inoltre di ottenere un dataset aggiornato di tutti gli orari e le giornate dei lavaggi stradali in qualsiasi posizione. L'inserimento spontaneo da parte degli utenti permette di realizzare una community responsabile e corretta per l'inserimento delle nuove entry nel data base real time. Non si è obbligati ad aggiungere nuove postazioni ma per poter attivare le notifiche al punto segnalato più vicino è consigliato aggiungerne uno corretto ancora più vicino se presente.

5. Ambiente di sviluppo e strumenti

L'ambiente di sviluppo integrato (IDE) utilizzato è Android Studio. Il progetto disponibile sulla repository pub-

blica [GitHub](#) è composto da tre principali file java che contengono il codice delle View:

1. MainActivity.java
2. MainActivity2.java
3. AlarmReceiver.java

I primi due sono essenziali per i file activity_main.xml e activity_main2.xml e contengono il codice per gestire la prima view ovvero della mappa con i vari marker scaricati dal data base real time Firebase Google con il rilevamento della posizione GPS e la seconda view gestisce l'inserimento di nuovi marker nel data base da parte dell'utente. La view principale permette inoltre di gestire ed attivare le notifiche di promemoria del lavaggio strade e si avvale della classe AlarmReceiver che permette di gestirle. Nel progetto sono presenti inoltre dei file gradle che permettono di definire le inclusioni di framework esterni come quello utilizzato per le mappe: MapBox.

5.1. MapBox

Le mappe implementate di [MapBox](#) sono open source e risultano una scelta vincente per lo sviluppo applicazioni Android. La modalità di visualizzazione può essere modificata dal codice e comprende varie tipologie tra cui: satellite, traffico, ibrida, notturna... La scelta più adatta è ricaduta infine su una mappa di tipo ibrido. MapBox inoltre permette di aggiungere dei marker su mappa passando le coordinate di latitudine e longitudine assieme ad un commento visualizzabile con un semplice tap sul marker. La mappa è orientabile in base alle esigenze ma la vista principale è in un'area vicino alla posizione attuale dell'utente assieme ai marker vicini.

5.2. Google Firebase

I marker sulla mappa vengono archiviati in un data base real time di [Google Firebase](#) che permette di tenere aggiornati i marker su tutte le mappe di qualsiasi utente. La gestione degli inserimenti e delle select è interamente affidata alla libreria fornita da Google che permette di operare sul database. Nella view della mappa principale vengono scaricati i marker nelle vicinanze e si tiene conto del più vicino tramite una semplice misura di distanza derivata dal teorema di Pitagora tra coordinate latitudine e longitudine. Nella vista del form da compilare invece, se i parametri risultano corretti e consoni ai campi inseriti si va ad aggiungere la entry nel database. Dunque le due view operano in modo differente ma analogo: una carica le entry se richiesto dall'utente, mentre l'altra le preleva per rappresentare i marker.

6. Diagramma di flusso

L'applicazione al primo avvio richiede i permessi da parte dell'utente per poter accedere alla posizione GPS del dispositivo Android e successivamente se accettati mostra la mappa nella posizione attuale con i marker. Infine è possibile attivare le notifiche o alternativamente se non è presente un marker nelle vicinanze si può segnalare aggiungendone uno al database real time andando a compilare il form.

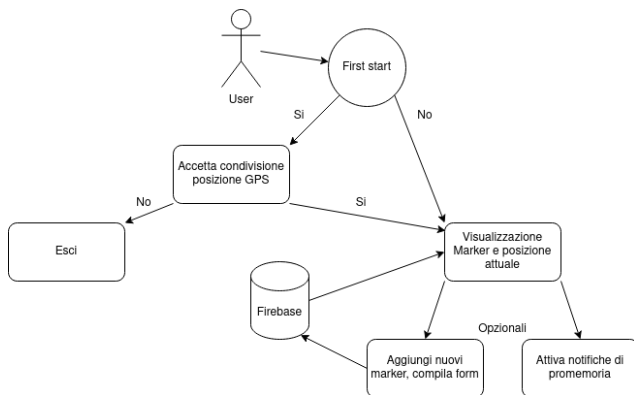


Figure 2. Flow Chart

Dal diagramma si evidenzia che la lettura del database avviene soltanto nella fase di posizionamento dei marker sulla mappa, mentre la scrittura solamente successivamente alla compilazione del form da parte di un utente.

7. Model View Controller

L'implementazione per una maggiore rigore è stata strutturata con riferimento al pattern Model View Controller (MVC) dividendo il codice in tre componenti:

1. Model: dati principali e modalità
2. View: mostra la mappa ed il form all'utente
3. Controller: gestisce gli input e media la comunicazione tra view e modello

Il codice risulta dunque più leggibile, manutenibile, adattabile grazie al fatto che il Software è diviso in parti più maneggevoli. In una applicazione Android si ha che il pattern è già implementato in maniera implicita poichè si hanno le corrispondenze tra: modello e dati, vista e file .xml, controllore e activity.

8. Views

L'applicazione sviluppata è stata concepita per essere il più semplice e intuitiva possibile, per questo motivo sono state realizzate due views principali. L'utente può sfruttare al

meglio l'applicazione anche tramite soltanto la prima view principale con mappa e marker. Si elencano successivamente le caratteristiche di ogni singola view.

8.1. Main View

Nella view principale, ovvero la view di inizio all'avvio dell'applicazione è stato implementato graficamente il mockup visto precedentemente con delle piccole variazioni che ne migliorano l'usabilità. L'elemento predominante è sicuramente la mappa con i relativi marker e la posizione gps, è possibile interagire con essi andando a vedere le etichette contenenti le informazioni dei relativi lavaggi nella posizione. Il pulsante "attiva promemoria" esprime al meglio ciò che compie, rispetta la buona norma per affordance e signifiers [1] andando appunto a prelevare la posizione del marker più vicino e inserisce un'allarme che tramite una notifica ricorderà di spostare la macchina. Quando il pulsante viene premuto si mostra inoltre un toast message che permette all'utente di capire di averlo premuto. È presente inoltre un altro pulsante che permette di accedere alla seconda view con il form da compilare se richiesto dall'utente.



Figure 3. Main View

8.2. Form View

Per accedere a questa view si deve aver premuto precedentemente il pulsante "Aggiungi lavaggio strade" che permette a tutti gli utenti, senza login o altre registrazioni, di aggiungere direttamente sul data base dell'applicazione nuovi marker per segnalare ad altri utenti i lavaggi. La vista del form è composta da switch e checkbox che permettono

all'utente di capire cosa seleziona e cosa no, il controllo dei dati inseriti come orario e selezione di almeno un giorno e di una settimana sono stati implementati nel controllo quando si preme il pulsante Submit. Nel caso di inserimenti errati il database non viene compromesso aggiungendo l'entry errata ma si mostra un messaggio toast all'utente dove viene richiesto di compilare correttamente e nel caso riguardare i campi inseriti. Per semplicità e un miglior design visivo si è scelto un layout a griglia di tipo responsive che si adatta su svariati dispositivi andando a variare solamente la dimensione dei pulsanti senza modificarne il collocamento.

The form titled "LavaggioStrade" contains the following elements:

- Days of the week: Lunedì, Martedì, Mercoledì, Giovedì, Venerdì, Sabato, Domenica. Martedì and Venerdì have active toggle switches.
- Frequency selection: ☐ Prima, ☐ Seconda, ☒ Terza, ☐ Quarta.
- Time selection: "Ora di inizio:" with a value of 14, and "Ora di fine:" with a value of 16.
- A purple "SUBMIT" button.
- A numeric keypad with digits 1-9, 0, a decimal point, a minus sign, a backspace key, and a confirmation key.

Figure 4. Form View

9. Scenari

Per capire quanto l'applicazione realizzata fosse user friendly si sono realizzati degli scenari da presentare alle persone intervistate in modo tale da astrarle dal fatto di essere sottoposte ad un test. Prima di effettuare un test con punteggio si è svolto un test preliminare non riportato nei punteggi in modo tale da comprenderne l'efficacia. Di seguito si riporta il testo di uno scenario:

1. Sentendo parlare da un amico, dopo svariate multe, dell'applicazione StreetCleaningMap di lavaggio stradale decidi di scaricarla e di aprirla.
2. Prova a localizzarti per la prima volta e guardare nelle tue vicinanze se vi è un marker che indica i giorni del lavaggio strade
3. Se per curiosità premi su "Aggiungi lavaggio strade" scopri le funzionalità offerte

4. Compila il form con le indicazioni del lavaggio a te più prossimo e premi "Submit"
5. Scorri la mappa e premi su "Attiva promemoria" per il più vicino marker
6. Supponi di spostare la macchina e di voler attivare le notifiche in una nuova posizione

Lo scenario riportato ha permesso dunque di valutare aspetti importanti dell'applicazione quali: semplicità d'uso, intuitività, apprendibilità e diletto. Per il punto 1 poichè l'applicazione non è stata pubblicata sul play store si è considerato passato nel momento in cui l'utente ha effettuato la ricerca per scaricarla (sfortunatamente la complessità di ricerca dell'applicazione non dipende dall'implementazione realizzata ma è universale per i dispositivi Android).

10. Usability Test

Le persone reclutate per il test sono state selezionate in ambito familiare tra i conoscenti e fuori da un bar (zona Gialla). In totale i partecipanti al test sono stati 15, tutti utilizzatori di auto e dunque con patente, in età compresa tra i 18 ed i 65 anni di entrambe i sessi. Nelle successive tabelle si riportano le distribuzioni dell'età e del sesso degli intervistati:

18-35	36-50	51-65
65%	28%	7%

Maschi	Femmine
57%	43%

Agli intervistati dopo aver svolto il test è stato dato successivamente un breve questionario con domande di tipo Single Ease Question (SEQ) in scala da 1 a 7 [2] per valutarne il livello di soddisfazione post task. Il questionario è stato formulato con un form di Google per poter successivamente estrapolare i dati ed i risultati in maniera efficace e rapida. Il tempo impiegato dagli intervistati è stato in media di 12 minuti ed era stato inizialmente provato in fase di test pilota. Le domande poste possono essere riassunte nella successiva tabella.

Domanda	Valor Medio	Varianza
Trovo semplice visualizzare la mia posizione	6.5	0.25
La posizione rilevata è corretta	6.8	0.1
Trovo comprensibili i pulsanti mostrati	6	0.3
Risulta difficile compilare l'aggiunta di lavaggi strade	5.2	0.45
Sono inutili i marker troppo lontani	2.3	0.8
La navigabilità dell'app non è chiara	1.8	0.3
I pulsanti non rispecchiano il comportamento	1	0
Mancano feedback dopo alcune interazioni	2.3	0.4
L'applicazione è noiosa da usare	2.1	0.3
L'applicazione è utile	6.5	0.4
Sono soddisfatto dell'app	6.6	0.2
Consiglierei l'app ad un amico	6.1	0.9
Preferirei una versione per iOS	4.2	1.4

Dai risultati evidenziati in fase di usability test si nota dunque che l'applicazione risulta ben sviluppata e a livello di esperienza utente agevola l'utilizzo. Alcuni utenti preferirebbero avere anche una versione per iOS, significa che le informazioni fornite dall'app risulterebbero interessanti anche per altri utenti. Poiché l'applicazione localizza la posizione automaticamente, all'utente basta solamente consentire il rilevamento GPS e successivamente attivare premendo un solo pulsante i promemoria. Questa scelta risulta vincente infatti il punteggio attribuito alla semplicità dell'utilizzo è importante e risulta un buon valore. Molti utenti consiglierebbero l'applicazione dunque emerge che l'applicazione è facile da utilizzare e con una buona velocità di apprendimento da parte dell'utente.



Figure 5. Alcuni intervistati in fase di usability test

11. Conclusioni

Il processo di sviluppo e la fase di usability test hanno permesso di realizzare una applicazione utile e gradevole all'utente che la scaricherebbe in futuro. Il progetto realizzato può essere successivamente espanso aggiungendo altri dettagli in futuro come numeri di parcheggi, indicazioni su parcheggi a pagamento e rispettivi orari ed altrettante idee. La progettazione e la fase finale di testing hanno evidenziato punti di forza e debolezze dell'applicazione in modo tale da essere migliorata in futuro con aggiornamenti sempre più utili all'utente finale. Le persone hanno trovato l'applicazione utile e divertente da usare, infine sono state numerose le battute sul fatto che grazie ad essa calerebbero drasticamente i numeri di multe.

References

- [1] A. D. Bagdanov. Human computer interaction, 2020. Figuring Stuff Out about ObjectsAffordances, Signifiers and Mapping, 02_affordances.pdf.
- [2] A. D. Bagdanov. Human computer interaction, 2020. Usability Testing, 27_usability.pdf.