

Report Laboratorio di applicazioni mobili - Tracking My Pantry

Michele Cucci
`michele.cucci@studio.unibo.it`

Università di Bologna, Laurea Triennale in Informatica

Gennaio 2022

Indice

1	Overview	3
2	Schermate principali	4
2.1	Schermata di login	4
2.1.1	Scelte implementative	5
2.2	Schermata principale	5
2.2.1	Schermata Home	6
2.2.2	Schermata Barcode Scanner	6
2.2.3	Schermata Settings	6
2.2.4	Scelte implementative	7
2.3	Schermata Inserimento Prodotto	8
2.3.1	Scelte implementative	8
2.4	Schermata dettagli prodotto	10
2.4.1	Scelte implementative	11
2.5	Notifiche	11
2.5.1	Scelte implementative	12

1 Overview

Food Guru è un'applicazione che permette all'utente di tenere traccia degli alimenti disponibili o da comprare per la propria dispensa. L'applicazione permette di trovare prodotti su un database condiviso tramite l'inserimento o la scansione di un codice a barre e di inserirne di nuovi nel caso non siano ancora presenti. L'applicazione inoltre dispone di un sistema di login per accedere, di visualizzare informazioni sui propri prodotti e di scansionare codici a barre tramite il lettore di codici a barre inserito all'interno dell'applicazione. Infine per ogni prodotto viene salvata le date di acquisto e di scadenza per poter monitorare e nel caso avvisare l'utente se un prodotto non è stato consumato in tempo. L'applicazione è stata sviluppata tramite l'IDE di sviluppo Android Studio con Java come linguaggio di programmazione.

Sì è scelto di utilizzare le seguenti librerie di supporto:

- Room

La libreria di persistenza Room fornisce un livello di astrazione su SQLite per consentire un accesso al database più solido. Ciò ha permesso un approccio più stabile e veloce al salvataggio dei dati sulla memoria.

- Camera X

CameraX è una libreria che fornisce una serie di correzioni per la compatibilità e soluzioni alternative per semplificare l'aggiunta di una fotocamera alla propria app. Ciò ha permesso l'utilizzo della fotocamera per eseguire scansioni di codici a barra.

- ML Kit

ML Kit è una libreria sull'intelligenza artificiale. È stata usata per implementare lo scanner dei codici a barra.

- Retrofit2

Retrofit è un client HTTP type-safe per Android e Java che trasforma le API HTTP in interfaccia Java. Ciò ha permesso una comunicazione tra client e server più semplice e sicura.

- Picasso

Picasso è una libreria di gestione/caricamento immagini con cui è stato possibile effettuare il caricamento delle immagini nella applicazione.

- WorkManager

WorkManager è una libreria utilizzata per la gestione di lavori persistenti con cui è stato possibile gestire il sistema di notifiche utilizzato.

2 Schermate principali

Per la realizzazione dell'applicazione si è scelto l'utilizzo di poche **Activity**, supportate da numerosi **Fragment**.

Di seguito sono mostrate le parti salienti dell'implementazione.

2.1 Schermata di login

La schermata d'accesso dell'applicazione è composta da due schermate: Login e Register.

La schermata con cui si interfaccia inizialmente l'utente è la schermata di login (Figura 1.a), l'utente può inserire le proprie credenziali per l'accesso e richiedere all'app di salvare i dati con cui auto-accedere le prossime volte.

La schermata di registrazione (Figura 1.b) permette ad una persona di creare un utente per poter accedere all'applicazione.

The image displays two mobile application screens side-by-side, each with a status bar at the top showing the time 10:23, signal strength, and 85% battery.

Screen (a) - Login: The title "Login" is centered at the top. Below it are two input fields labeled "Email" and "Password". A red "SIGN IN" button is positioned below the password field. Underneath the button are two checkboxes: "Remember the password" and "auto login". At the bottom, a right-pointing chevron icon is accompanied by the text "Swipe left for register".

Screen (b) - Register: The title "Register" is centered at the top. Below it are four input fields labeled "Username", "Email", "Password", and "Re-type Password". A red "REGISTER" button is located below the "Re-type Password" field. At the bottom, a left-pointing chevron icon is accompanied by the text "Swipe right for login".

(a) schermata di login

(b) schermata di registrazione

Figura 1: Le schermate d'accesso dell'app

2.1.1 Scelte implementative

Per l'implementazione delle schermate d'accesso si è scelto di implementare nell'**Activity** un **ViewPager2** che permette all'utente di scorrere fra i due **Fragment** in maniera agevole e fluida. Per eseguire le richieste HTTP si è deciso di utilizzare due classi java apposite tramite **Retrofit2**. Infine l'implementazione della funzione auto-login sfrutta il salvataggio l'username e la password dell'utente in un **SharedPreferences**.

2.2 Schermata principale

Dopo aver eseguito il login, l'utente accede alla schermata principale.

La schermata principale è composta da tre sotto-schermate: la schermata **Home** (dove si trova l'utente), la schermata **Barcode Scanner** e la schermata **Settings**.

Per muoversi tra le tre sotto-schermate viene utilizzata una barra di navigazione.

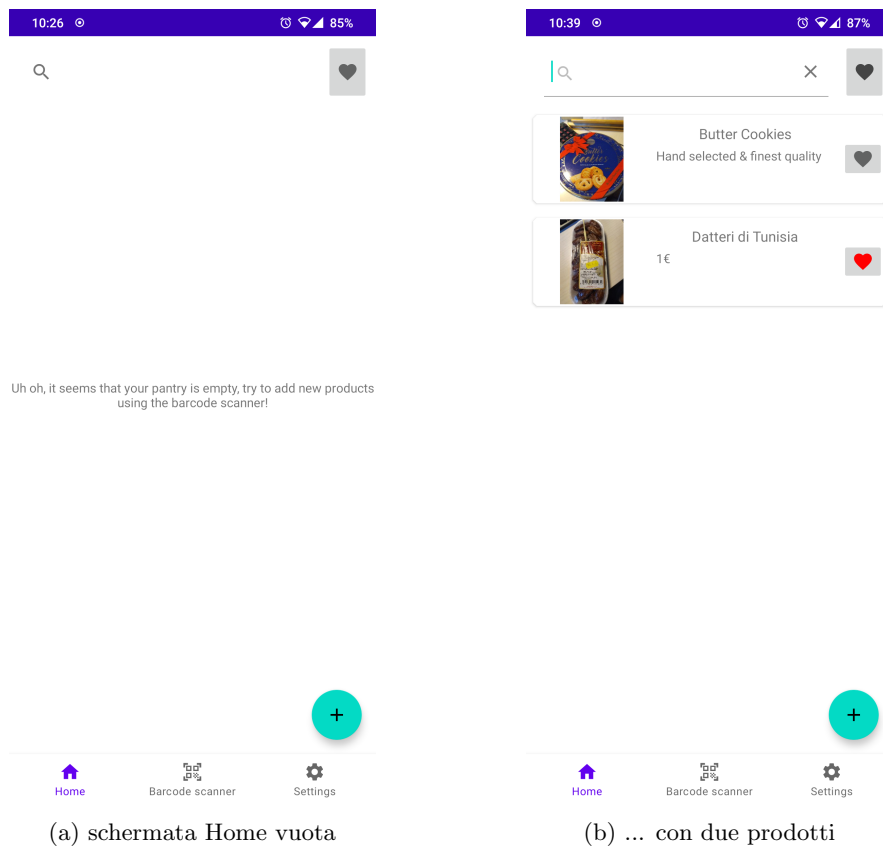


Figura 2: schermata Home

2.2.1 Schermata Home

La schermata Home (Figura 2) è il cuore dell'applicazione, cioè dove l'utente trova i vari prodotti che ha inserito nel database locale.

I prodotti sono visualizzabili tramite l'utilizzo di **Cards** che mostrano il nome, la descrizione e l'immagine del prodotto; inoltre ogni **card** ha un bottone per inserire il prodotto tra i preferiti.

Infine è possibile effettuare una ricerca sul database locale in due modi: inserendo il nome del singolo prodotto che si sta cercando nella barra di ricerca o premendo il pulsante ad essa adiacente per visualizzare tutti i prodotti preferiti.

2.2.2 Schermata Barcode Scanner

La schermata Barcode Scanner permette all'utente di utilizzare la propria videocamera posteriore per scannerizzare un codice a barre di un prodotto che vuole inserire. La prima volta che viene eseguita questa funzionalità verranno richiesti all'utente i permessi per l'utilizzo della fotocamera. Solo in caso vengano vengano acconsentiti verrà accesa la fotocamera e mostrata a video finché non scannerizzerà un codice.

2.2.3 Schermata Settings

Per concludere abbiamo la schermata Settings (Figura 3) che contiene le varie impostazioni: abilitazione delle notifiche, impostazione di un limite di tempo per il controllo di vita nei prodotti, logout dell'utente dall'applicazione.

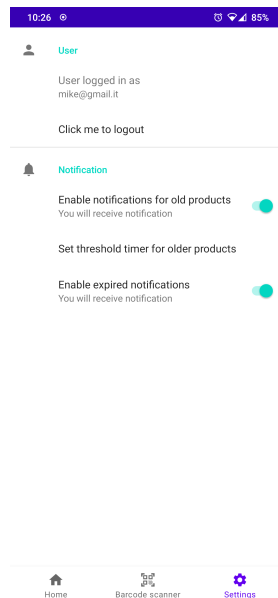


Figura 3: schermata Settings

2.2.4 Scelte implementative

- Schermata Principale

Per la realizzazione della schermata principale si è scelto di applicare all'Activity **MainActivity** un barra di navigazione **NavController** che permette di navigare in uno dei tre Fragments che compongono il corpo dell'applicazione diviso in tre sotto-schermate principali.

- Schermata Home

L'implementazione della schermata **Home** ricorre all'utilizzo di una **RecyclerView** contenente delle **Cards** gestite tramite **Adapter**. Ogni carta visualizza i valori di uno dei prodotti contenuti all'interno del database locale.

Tramite la **SearchView** presente nella parte alta è possibile eseguire ricerche applicando un filtro alla **Repository** dove sono contenuti i dati: il risultato è la rimozione dalla **RecyclerView** di tutti gli elementi che non rispettano il filtro. Inoltre è possibile visualizzare solamente gli elementi preferiti premendo l'**ImageButton** a forma di cuore accanto alla **SearchView**, che applicherà una specifica query SQL alla **RecyclerView** per visualizzare solamente gli elementi preferiti.

Infine premendo il **FloatingActionButton** con un "+" al suo interno compare una **DialogFragment** che permette di ricercare un prodotto tramite un codice a barre.

- Schermata Barcode Scanner

La schermata è composta interamente da una **PreviewView** che non mostra niente finché non gli vengono conferite le autorizzazioni necessarie. Appena viene attivata la schermata, la funzione **ImageAnalysis** si inserisce sul **MainExecutor** ed inizia ad analizzare lo stream di frame fornito dalla fotocamera, finché non trova un codice a barre. Appena riconosce un codice a barre come reale, viene inviato tramite al metodo **Intent** alla schermata per inserimento prodotti.

- Schermata Setting

Le impostazioni sono implementate tramite l'uso della libreria **Settings** di Jetpack, che permette di memorizzare impostazioni come **Preferences**. Ciò permette all'utente di rimuovere notifiche non desiderate ed inoltre fornisce la possibilità di effettuare log-out eliminando gli **SharedPreferences** corrispondenti.

2.3 Schermata Inserimento Prodotto

Quando viene inserito un codice a barre per effettuare l'inserimento di un nuovo prodotto, l'applicazione visualizza la schermata per inserimento del prodotto. Essa è composta da due sotto-schermate che mostrano rispettivamente i prodotti presenti nel database remoto (Figura 4.a) ed una schermata apposita per consentire l'inserimento di un nuovo prodotto (Figura 4.b), nel caso in cui non sia stato trovato il prodotto cercato.

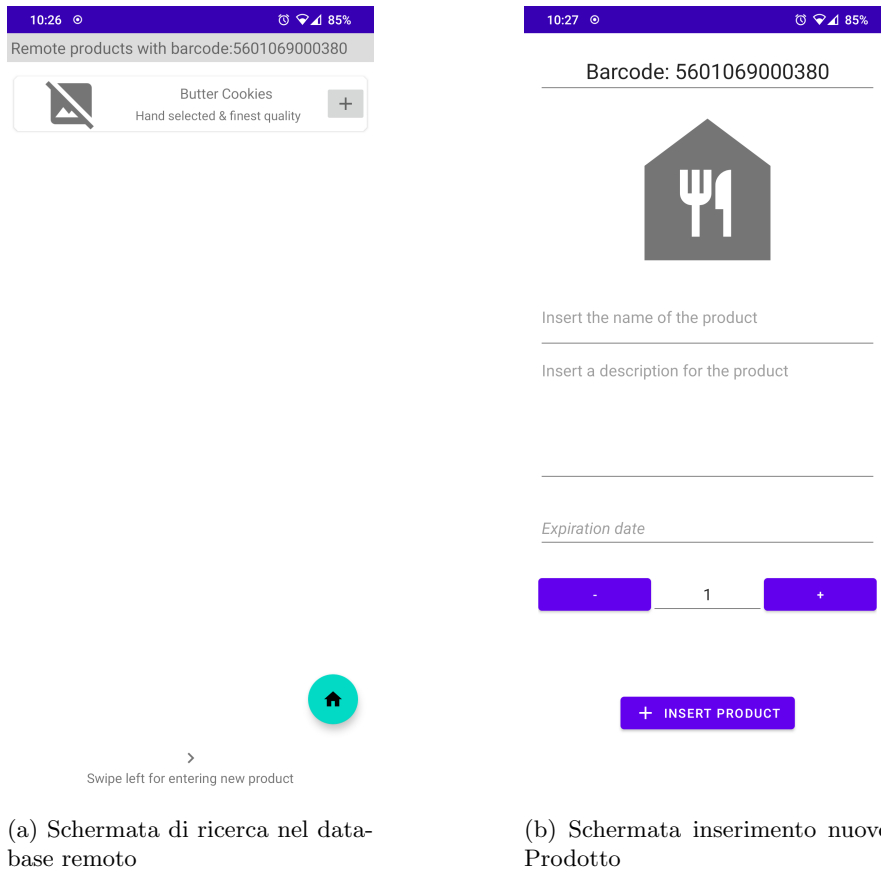


Figura 4: Schermata Inserimento Prodotto

2.3.1 Scelte implementative

Per l'implementazione della **Schermata Inserimento Prodotto** si è scelto di utilizzare la stessa logica applicata per la schermata login, ossia un **Activity** principale contenente un **ViewPager2** che permette all'utente di scorrere tra due frammenti.

Il primo **Fragment** (Figura 4.a) è costituito da una **TextView** che mostra il

barcode per cui si sta effettuando la ricerca, un **FloatingActionButton** che ritorna alla schermata principale richiamando un Intent e termina la schermata attuale ed una **RecyclerView** che mostra una Card per ogni prodotto con lo stesso barcode sul server condiviso e non attualmente presente all'interno del database locale.

Ogni Card contiene, oltre alle informazioni principali del prodotto, un **ImageButton** che permette l'aggiunta del prodotto al database locale tramite query SQL **Update** sul **Repository** del DB.

Scorrendo a destra ci troviamo nella schermata di inserimento prodotto (Figura 4.b), che permette l'inserimento di un prodotto tramite la compilazione di **TextView** per le informazioni basi e un **ImageView** per l'inserimento di un'immagine, potendo scegliere tra camera o galleria attraverso un DialogFragment **SelectPhoneDialogFragment**.

L'inserimento del prodotto nei DB locale e in quello remoto è eseguito tramite una chiamata Rest al Server. Nel caso la chiamata abbia avuto successo il prodotto viene inserito nel DB locale tramite la query SQL **Insert** e la schermata attuale viene chiusa facendo ripartire la schermata principale.

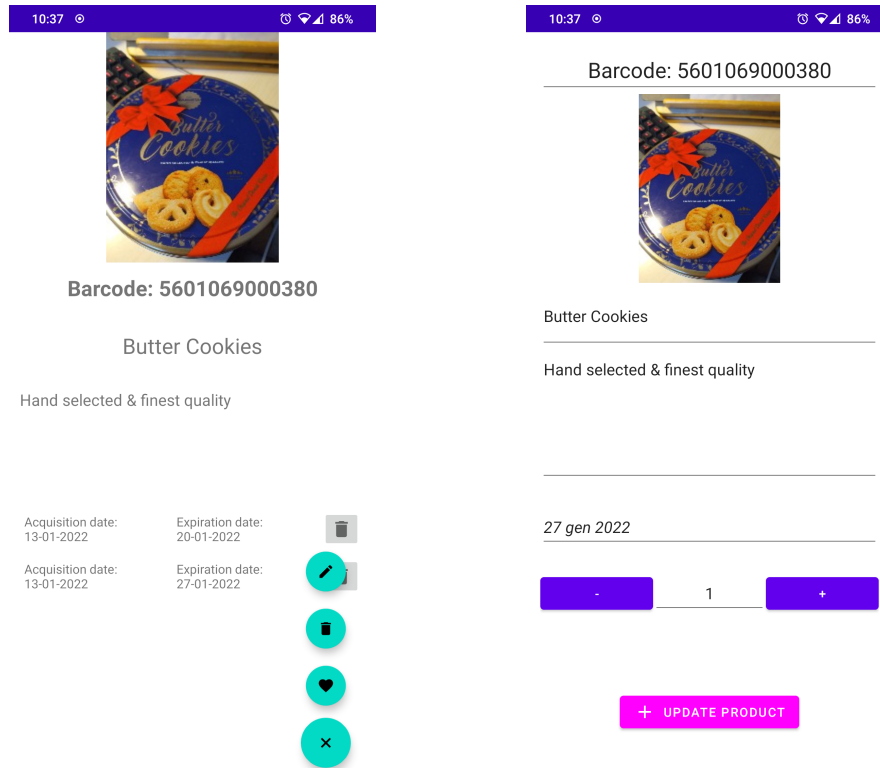
2.4 Schermata dettagli prodotto

Premendo una delle carte nella schermata Home si potrà accedere alla schermata relativa al singolo prodotto. (Figura 5.a)

Nella schermata verranno mostrate tutte le informazioni generali: codice a barra, nome, descrizione ed immagine.

Sono visualizzate anche le varie "istanze" del prodotto, ognuna con data d'acquisto e di scadenza. Ogni istanza è cancellabile utilizzando il bottone a forma di bidone presente in ogni Card.

Inoltre premendo il bottone "+" visibile nella parte bassa della schermata è possibile accedere ad un sotto menù di azioni disponibili: eliminare il prodotto dal database, aggiungere il prodotto agli elementi preferiti e modificare il prodotto. La modifica del prodotto utilizza la stessa schermata di quella di inserimento del prodotto (Figura 5.b), ma modificata con tutti i dati disponibili e con l'azione da inserimento al posto dell'azione di modifica.



(a) schermata contenente tutti i dettagli del prodotto

(b) schermata di modifica del prodotto

Figura 5: schermata dettagli prodotto

2.4.1 Scelte implementative

Premendo una Card nell'Activity Home viene chiamato un **Intent** che trasporta i dati del prodotto, associato alla Card, all'Activity **FullProductCard** che mostrerà i dati nelle varie TextView.

Le varie "istanze" del prodotto sono visualizzate tramite un RecyclerView.

Nel momento in cui si seleziona il bottone per la modifica del prodotto, vengono passati i dati alla schermata di Inserimento Prodotto, attivata tramite un Intent contenente il prodotto compresso in un container **Parcelable**. Aperta la nuova schermata vengono mostrati i dati nelle TextView rispettive e viene modificato il funzionamento della pagina in modo che esegua solamente una query SQL **Update** e non invii nessun nuovo dato al server remoto.

2.5 Notifiche

L'applicazione invia notifiche ogni giorno riguardo i prodotti disponibili. Le notifiche riguardano i prodotti che hanno raggiunto la loro data di scadenza e i prodotti che sono diventati vecchi e devono essere consumati il prima possibile. Interagendo con le notifiche l'applicazione si aprirà nella schermata di login per permettere all'utente di accedere al database. Per i dispositivi con versione di Android API superiore alla 26 vengono aggiunti specifici canali per la trasmissione delle notifiche.

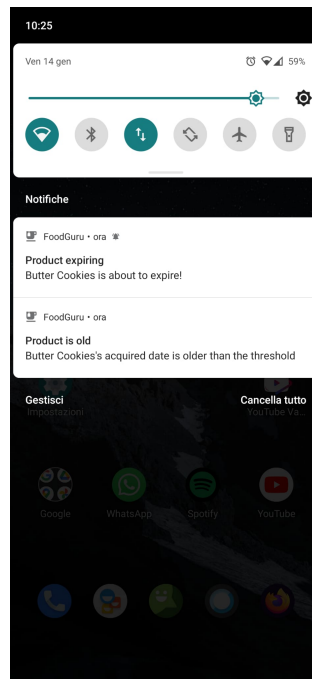


Figura 6: schermata dettagli prodotto

2.5.1 Scelte implementative

All'avvio l'applicazione esegue un override dello start-up standard tramite la classe App.

Nella classe App viene inizializzata una istanza di **WorkManager**; successivamente vengono lanciate due **PeriodicWorkRequests** che ciclano giornalmente sulle classi **NotificationExpiredWorker** e **NotificationOldWorker**. Inoltre nella classe App vengono inizializzati i canali per le notifiche se il livello delle Android API è adeguato.

Le due classi eseguono l'override del metodo **doWork**. Ciò permette di controllare se ci sono prodotti in scadenza o con date d'acquisto troppo vecchie ed eventualmente di lanciare la corrispondente notifica.