

Documentazione YourGeekFriend



Nome e Logo dell'applicazione

Per definire il nome ci siamo rifatti a quella che è la natura della nostra applicazione, che è stata pensata come un' enciclopedia che fornisce informazioni su oggetti riconosciuti in tempo reale.

'YourGeekFriend' tradotto in italiano significherebbe letteralmente il tuo amico "secchione", la nostra idea era proprio quella di richiamare la figura dell' amico "secchione" che riesce a rispondere a tutte le tue possibili domande.

'YourGeekFriend' è quindi il tuo piccolo amico virtuale, che attraverso il suo unico e preciso "occhio", riesce a fornirti informazioni su una vasta gamma di oggetti, che gli vengono mostrati.

Il logo dell'applicazione invece è ispirato ai baffi di uno dei più grandi geni della storia dell'umanità, Albert Einstein.

Descrizione generale

Si tratta di un'applicazione mobile, che sfrutta la fotocamera del tuo dispositivo e un API, per l'intelligenza artificiale, per riconoscere gli oggetti inquadrati e fornire all'utente informazioni riguardanti questi oggetti.

L'obiettivo è quello di fornire agli utenti una vera e propria enciclopedia in grado di fornirgli, in tempo reale, informazioni sull'oggetto inquadrato e la possibilità di cercare ulteriori informazioni su quest'ultimo, mediante il motore di ricerca di Google e/o la sua pagina Wikipedia.

L'obiettivo è quindi quello di rendere il tutto molto immediato.

Requisiti e compatibilità

L'applicazione è compatibile con i dispositivi dotati di un sistema operativo Android, nella versione 5.0 (Lollipop) o superiore, o iOS nella versione 11.0 o superiore.

Documentazione tecnica

Strumenti utilizzati per lo sviluppo

Per lo sviluppo della versione Android sono stati utilizzati l'IDE Android Studio, un dispositivo Android per l'emulazione e Tensorflow, un API open source di proprietà di Google, che consente agli sviluppatori di sfruttare degli algoritmi di machine learning su un singolo dispositivo.

Per lo sviluppo della versione iOS invece sono stati utilizzati l'IDE Xcode, un iPhone per l'emulazione e l'API di computer vision di Apple.

Per la gestione dello storage invece, entrambe le versioni utilizzano un dbms noSql, in cloud, di proprietà anche questo di Google, Firebase.

Versione iOS

Apple Vision

Nello sviluppo della versione iOS dell'applicazione, è stata utilizzata un'API di proprietà Apple di nome Vision.

Il Framework Vision offre diverse funzionalità: permette il riconoscimento dei volti, rilevamento del testo, riconoscimento di codici a barre. Inoltre consente l'utilizzo di modelli personalizzati di intelligenza artificiale (modelli Core ML), basati su Machine Learning, che consentono di svolgere attività quali la classificazione e/o il rilevamento di oggetti in tempo reale.

Utilizzando il Core ML framework, la libreria Vision sarà in grado di sfruttare i modelli basati sulle reti neurali per classificare i dati forniti in input. Visualizzata l'immagine tramite la fotocamera del dispositivo, questa verrà processata ed elaborata per classificare l'immagine ottenuta.

Creando il progetto in Xcode, Vision automaticamente genererà una classe in Swift che consentirà facilmente di accedere al modello ML. Per il progetto in questione il modello ML utilizzato è Inception v3, dunque automaticamente tramite la collaborazione tra il framework Vision e il framework Core ML verrà generata la classe InceptionV3 derivata dal modello ML importato.

Successivamente per elaborare il contenuto dell'immagine, basato sul modello precedentemente creato, sarà necessario creare un'istanza della classe VNCoreMLRequest.

Di seguito è riportata la porzione di codice precedentemente descritta.

```
let model = try VNCoreMLModel(for: InceptionV3().model)

let request = VNCoreMLRequest(model: model, completionHandler: { [weak self]
request, error in
    self?.processClassifications(for: request, error: error)
})
return request
```

CaptureDevice

Per poter accedere alla fotocamera del dispositivo iOS, è necessario secondo le normative della privacy previste, chiedere il permesso all'utente. all'interno del codice successivamente è necessario importare la libreria AVKit che consente di gestire il flusso dati della fotocamera.

```
import Vision
```

All'interno del metodo ViewDidLoad(), che viene richiamato subito dopo l'apertura dell'applicazione, viene inizializzato l'oggetto chiamato captureSession per la manipolazione del flusso dati della fotocamera, tramite il metodo AVCaptureSession().

```
let captureSession = AVCaptureSession()
```

Si prosegue assegnando alla variabile captureDevice come modalità predefinita della fotocamera la modalità video e infine viene avviata la fotocamera tramite il metodo startRunning().

```
guard let captureDevice = AVCaptureDevice.default(for: .video) else { return }
```

```
guard let input = try? AVCaptureDeviceInput(device: captureDevice) else { return }
```

```
captureSession.addInput(input)
```

```
captureSession.startRunning()
```

Versione Android

TensorFlow

Nello sviluppo della versione Android, come accennato in precedenza, è stata usata un API open source di Google, Tensorflow, nella sua versione Lite, una versione creata appositamente per il mondo mobile.

La libreria offerta da TensorFlow consente l'interazione con dei modelli di intelligenza artificiale, basati sulle reti neurali.

Il tutto infatti avviene all'interno del dispositivo, senza il bisogno di dover sfruttare la potenza di calcolo di ulteriori sistemi back end, a cui ci si dovrebbe altrimenti connettere attraverso richieste http, rallentando così i singoli riconoscimenti.

Camera preview

Per prima cosa, per poter utilizzare la fotocamera di un dispositivo Android, è necessario dichiarare questo intento nel Manifest dell'applicazione, andrà inoltre previsto del codice per la richiesta dei permessi necessari e per il controllo della loro effettiva concessione.

Per la gestione della fotocamera, è stata utilizzata la nuova Camera2Api inclusa nell'sdk Android ed utilizzabile dalla versione 5.0 del sistema operativo Android (Lollipop).

Questo package, offre una serie di classi utili per l'interazione con la fotocamera del dispositivo.

Per ottenere il controllo della fotocamera del dispositivo è stata utilizzata la classe CameraManager, che istanziata e connessa ad un oggetto di tipo CameraDevice, ci permette di ottenere un riferimento relativo ad una singola fotocamera connessa al dispositivo (un dispositivo ovviamente potrebbe essere dotato di più di una fotocamera, per questo ciascuna fotocamera viene identificata univocamente da un ID).

Ottenuto il riferimento al CameraDevice desiderato, che nel nostro caso è la fotocamera posteriore del dispositivo, siamo ora in grado di utilizzarne il metodo createCaptureSession per ottenerne l'anteprima, il metodo in questione richiede due parametri, il tipo di output desiderato, nel nostro caso TEMPLATE_PREVIEW, e l'oggetto beneficiario dell'output, nel nostro caso una TextureView.

La TextureView è un particolare oggetto grafico dell'ambiente Android, che viene utilizzato per mostrare in output all'utente contenuti grafici di tipo immagine o video, nel nostro caso sarà destinataria dei singoli frame provenienti dalla nostra fotocamera.

Classificazione frame camera preview

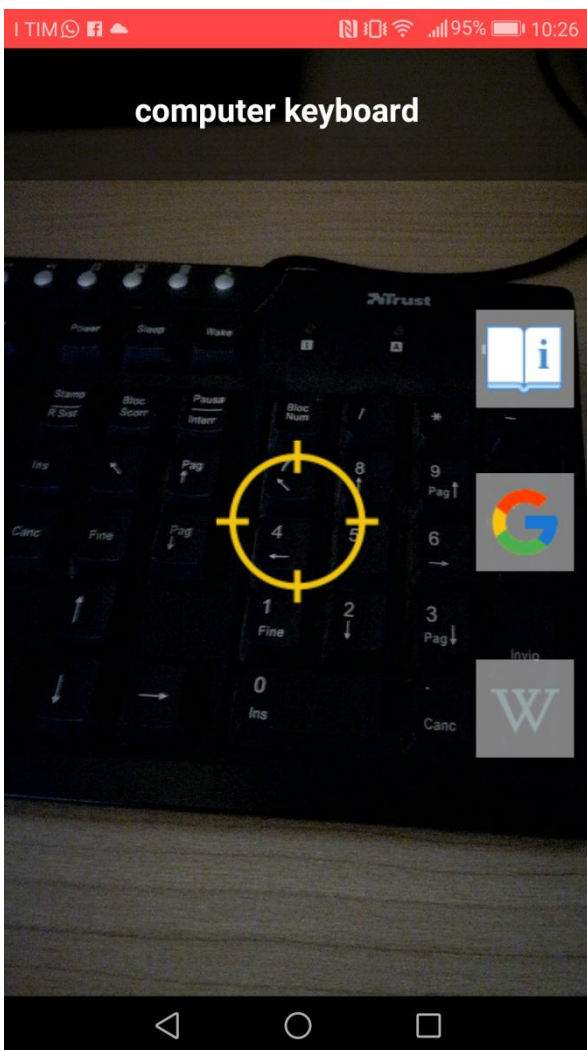
Il primo passaggio consiste nella conversione del frame in formato Bitmap.

Una volta convertito il frame viene passato come parametro al metodo `classifyFrame` dell'oggetto `ImageClassifier`, istanza di una classe incaricata al caricamento dei file relativi al modello di intelligenza artificiale e al file di testo con le relative etichette e alla chiamata dei metodi della libreria di TensorFlow.

Il metodo `classifyFrame`, restituisce l'etichetta relativa all'oggetto riconosciuto.

L'etichetta, che altro non è che una stringa, rappresentante il nome dell'elemento riconosciuto, viene assegnata alla `TextView`.

UI - Interfaccia utente



L'interfaccia utente è stata realizzata posizionando nella Main activity (la finestra principale dell'applicazione e anche l'unica) una `TextureView`, la cui funzione è stata descritta nei paragrafi precedenti dell'applicazione e consiste nell'“ospitare” la preview della fotocamera.

Sulla `TextureView` è stata posizionata una `TextView`, una casella di testo, che ha invece la funzione di mostrare i risultati di ciascun riconoscimento.

Oltre alla `TextView` sulla `TextureView` è stato posizionato un mirino grafico, che ha la sola funzione di supportare l'utente nell'inquadratura degli oggetti.

Inoltre, sempre sulla `TextureView` son stati posizionati 3 `ImageButton`, dei pulsanti con logo.

Il primo pulsante permette all'utente di visualizzare una descrizione “rapida” dell'elemento inquadrato, a patto che questo si trovi nel database.

Il secondo pulsante offre all'utente la possibilità di visitare la pagina Wikipedia relativa all'oggetto inquadrato.

Il terzo pulsante invece consente all'utente di eseguire una ricerca istantanea dell'elemento inquadrato, tramite il suo browser predefinito.

Firestore

Firestore è un servizio Web, cross platform (disponibile per applicazioni Android, iOS e Web) che tra le varie funzionalità, offre un servizio DBMS di tipo noSQL in cloud.

La nostra scelta è ricaduta su questo servizio perché ci permette la creazione di un database comune a tutti gli utenti e comune ad entrambi le versioni dell'applicazione (Android e iOS).

Il database, essendo in cloud, risulta quindi sempre accessibile, a patto che l'utente sia dotato di una connessione internet.

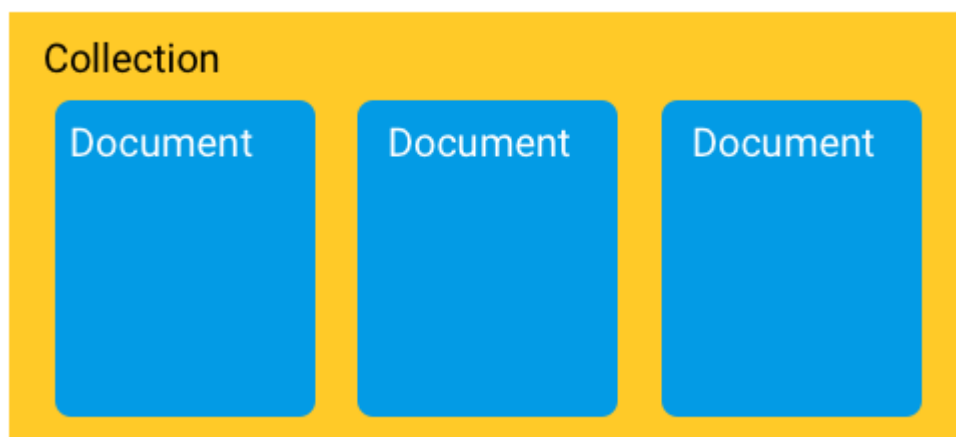
Il modello usato da Firestore si differenzia da quello usato dai DBMS tradizionali (di tipo sql) e non prevede la presenza di tabelle.

Il modello utilizzato da Firestore si basa sui concetti di documento (Document) e collezione (Collections).

Un documento, è un singolo elemento del database, che contiene vari campi, individuati da una coppia chiave-valore, ciascun documento viene identificato univocamente da una sua chiave primaria.

La collezione invece è definita come un contenitore di documenti.

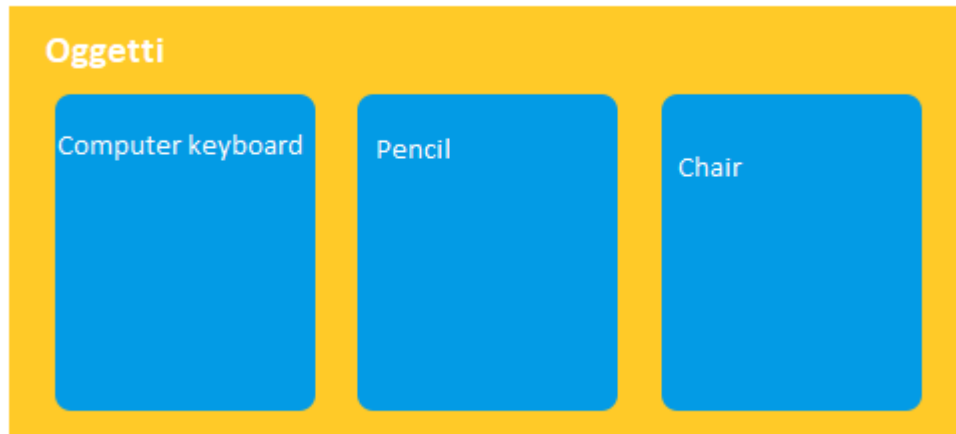
Per fare un'analogia con i database relazionali (sql) possiamo paragonare i documenti alle tuple di una tabella e le collezioni ad una tabella del database.



Il nostro database è costituito da un'unica Collection, la Collection 'Oggetti' che a sua volta contiene i vari Documents relativi agli oggetti presenti nel database.

Ciascun Document possiede un campo contenente il link alla pagina Wikipedia dell'oggetto e un campo contenente la sua descrizione.

Per comodità abbiamo scelto di far coincidere la chiave primaria di questi Documents con il nome dell'oggetto che rappresentano.



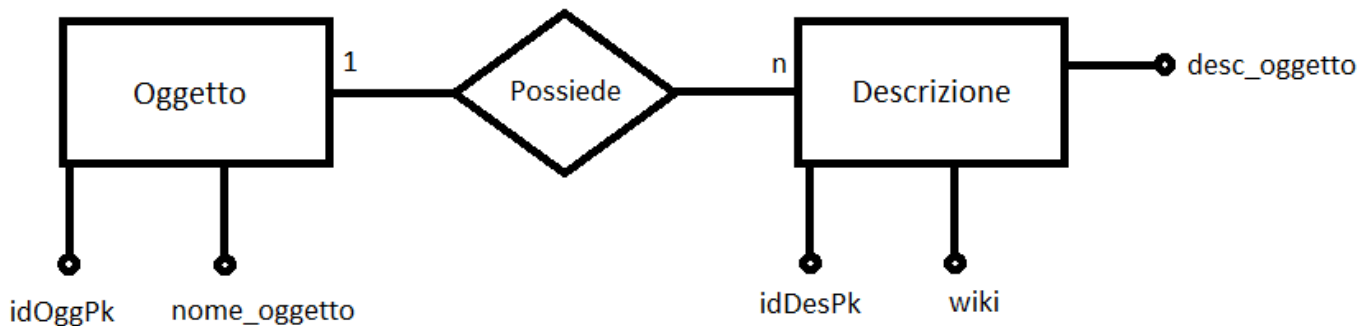
Collegamento tra applicazione e Database

Il collegamento tra DataBase e applicazione avviene tramite l'utilizzo di una chiave univoca fornita in fase di progettazione direttamente da Firebase. Questa chiave dovrà, in fase di programmazione, essere inserita all'interno dei rispettivi progetti (Android e iOS) in modo tale da consentire il riconoscimento da parte di Firebase delle rispettive applicazioni. Una volta concluso il collegamento importando la libreria Firebase-Firestore sarà possibile procedere con la manipolazione dei dati.

Database SQL

Il DataBase è lo strumento che ci permette di mantenere e gestire i dati ed è stato sviluppato nel seguente modo:

Modello E/R



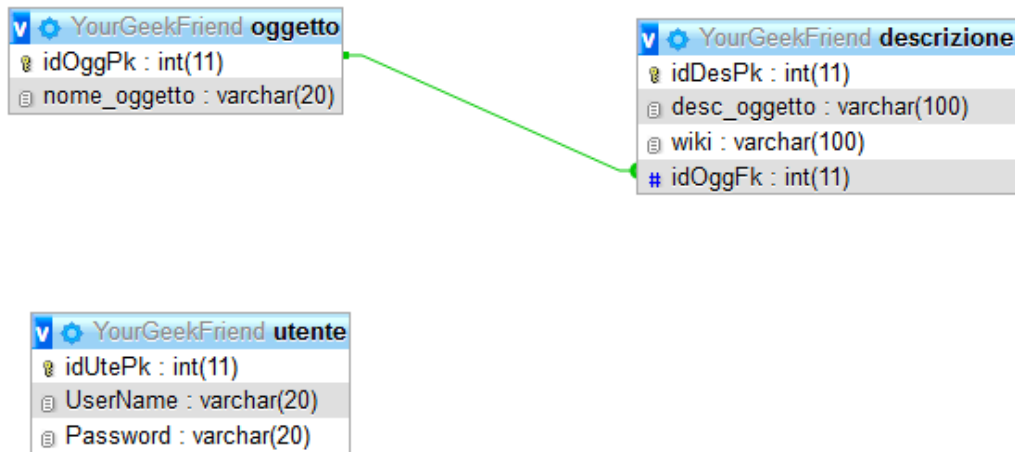
Nel db troviamo due entità: Oggetto e Descrizione. L'associazione tra di loro è di tipo uno a molti, in quanto un oggetto possiede una descrizione, mentre una descrizione può essere posseduta da uno o più oggetti.

L'entità oggetto possiede due attributi: la chiave primaria (idOggPk che identifica univocamente l'oggetto) e il nome dell'oggetto (nome_oggetto); l'entità descrizione possiede, invece, quattro attributi: la chiave primaria (idDesPk), la descrizione (descrizione), il link a Wikipedia (wiki) e una chiave esterna da Oggetto (idOggFk) che viene utilizzata per associare le due relazioni.

Modello logico

Tabella	Campo	Chiave	Tipo	Dimensione	Decimale
Oggetto	idOggPk	Primaria	int Auto_increment	6	//
	nome_oggetto	//	varchar	20	//
Descrizione	idDesPk	Primaria	int Auto_increment	6	//
	desc_oggetto	//	varchar	1000	//
	wiki	//	varchar	1000	//
	idOggFk	Esterna da Oggetto	int	6	//

DataBase relazionale



In quanto non abbiamo un server sul quale far girare il DataBase abbiamo optato per l'utilizzo di Firebase, un servizio di immagazzinamento dati online e facilmente gestibile da smartphone.

Query esempio (JOIN per ricevere la descrizione)

```
SELECT desc_oggetto
```

```
FROM Descrizione, Oggetto
```

```
WHERE nome_oggetto = '$nome_oggetto'
```

```
AND idOggPk = idOggFk
```

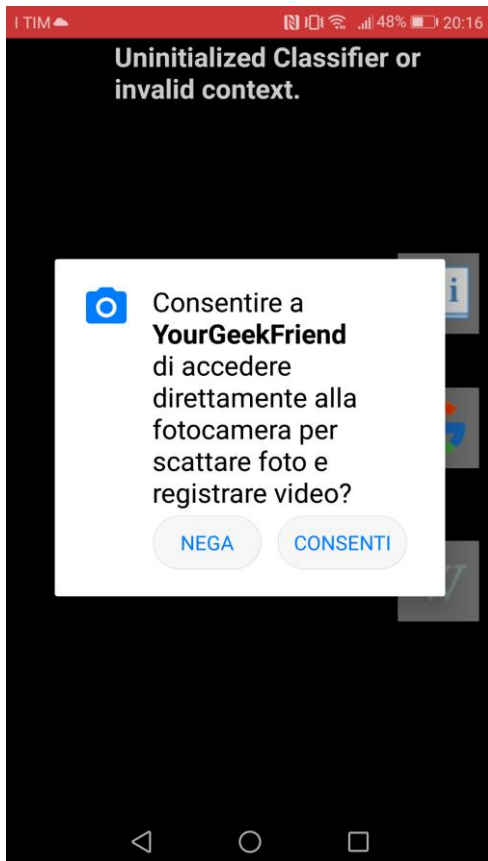
Documentazione Utente

Installazione

Una volta terminata, l'applicazione verrà caricata sugli store digitali e sarà disponibile al download su tutti i modelli compatibili.

L'utente dovrà quindi recarsi nel Google Play store, per scaricare l'applicazione sul suo dispositivo Android, e nell'Apple App store per scaricare l'applicazione sul suo iPhone.

Primo avvio



Al primo avvio dell'applicazione all'utente verrà richiesta la concessione dei permessi relativi all'uso della fotocamera, necessario alle funzionalità dell'applicazione.

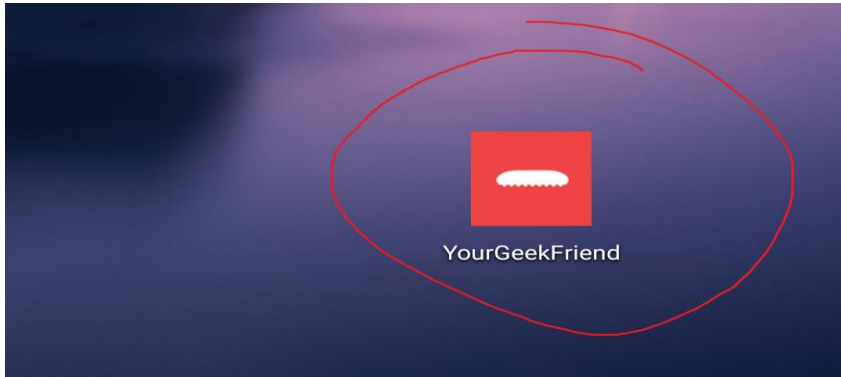
Nel caso l'utente neghi i suddetti permessi, non potrà procedere nell'utilizzo dell'applicazione, in quanto la fotocamera svolge un ruolo fondamentale per il suo funzionamento.

Utilizzo

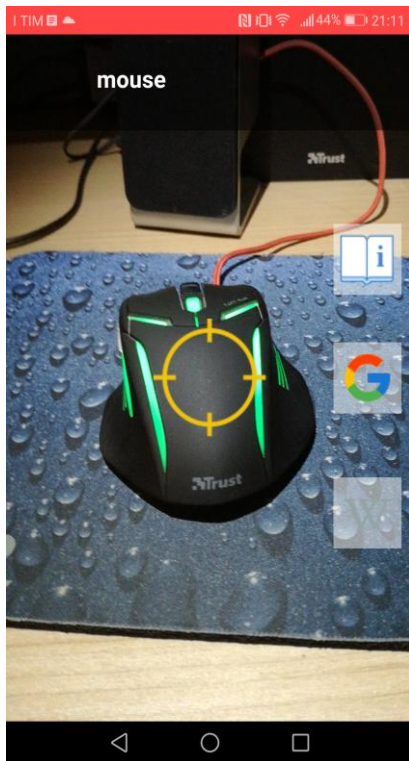
Concessi i permessi relativi alla fotocamera, all'utente non resta che utilizzare l'applicazione, che fa della sua semplicità di utilizzo e della sua immediatezza i suoi due principali punti di forza.

L'utente potrà sfruttare le funzionalità della nostra applicazione in **X** semplici step:

1. Aprire l'applicazione

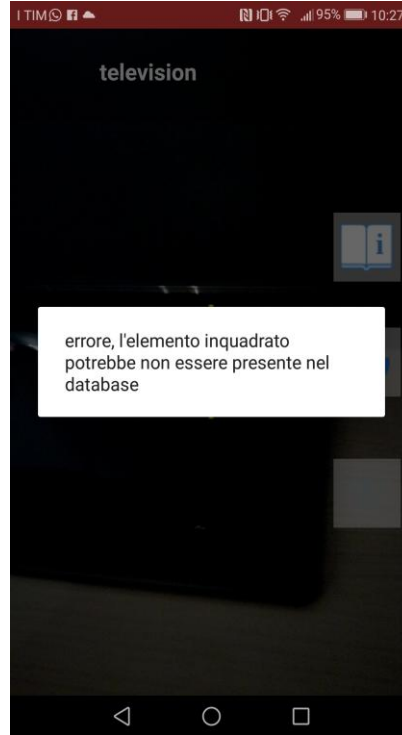
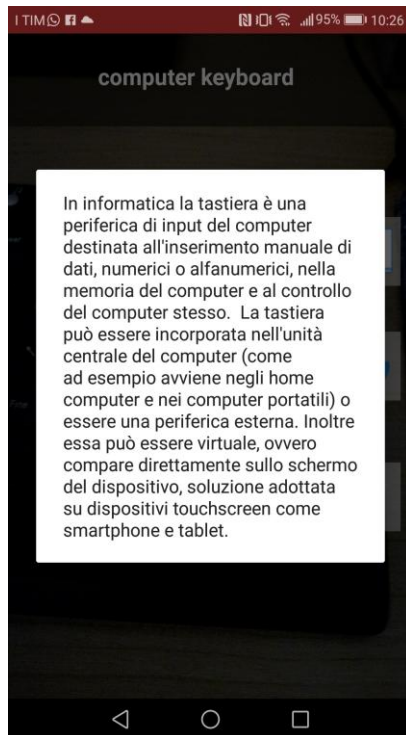


2. Puntare la fotocamera del dispositivo verso l'oggetto su cui si vuole ottenere informazioni, aiutandosi con il mirino grafico disposto al centro dello schermo



3. A questo punto, l'utente si trova di fronte a tre possibilità:

- 1) Premendo il primo pulsante, quello rappresentato da un'immagine raffigurante un libro, l'utente potrà visualizzare, se disponibile nel database, una descrizione "rapida" dell'elemento inquadrato, attraverso un alert che apparirà al centro dello schermo. In caso contrario all'utente verrà mostrato un alert che comunicherà la mancanza della descrizione per l'elemento inquadrato.



- 2) Premendo il secondo pulsante, quello con il logo di Google, verrà avviata una ricerca automatica, attraverso il browser predefinito del dispositivo, che avrà per oggetto l'elemento inquadrato.



- 3) Premendo il terzo pulsante invece, rappresentato dall'icona di Wikipedia, l'utente potrà visualizzare la pagina relativa all'oggetto inquadrato.



Analisi dei possibili contesti

L'applicazione inoltre, considerata la sua natura, potrebbe facilmente essere adattata ad altri contesti quali: quello farmaceutico, in cui, con le dovute modifiche, potrebbe essere utilizzata per il riconoscimento in tempo reale dei medicinali e per la ricerca immediata dei suoi generici e in quello alimentare, dove potrebbe essere utilizzata per esempio dal personale e/o dai clienti di un supermercato, per ottenere in tempo reale la lista degli ingredienti che compongono un qualsiasi prodotto inquadrato con la fotocamera del dispositivo.