

# Simple Sample

Applicazione per la registrazione di suoni ambientali e archivio digitale

# Idea

- Astrazione di un sistema Akai MPC: campionatore nato negli anni '80 che ha portato alla nascita vari generi musicali, quali HipHop e Elettronica, e che permette la registrazione di suoni, storicamente da vinili, insieme alla loro manipolazione in pitch e durata. Nasce quindi un “Sample”, il quale viene associato a uno dei 16 tasti per la riproduzione di cui è dotata la macchina.
- La mappatura sulla griglia permette la riproduzione arbitraria di tali suoni e la successiva creazione di arrangiamenti e basi musicali (il classico “Bum-Bap” tipico delle tracce HipHop è nato in questo modo).
- Simple Sample si basa sullo stesso concetto, permettendo di associare a ognuno dei 16 tasti mostrati un suono registrato tramite il microfono del dispositivo, e la successiva riproduzione dello stesso.
- L'applicazione viene inoltre fornita di un “Sequencer”, sistema di esecuzione automatica che permette di combinare la riproduzione di suoni associati ai vari tasti

# Features

Simple Sample offre le seguenti funzionalità:

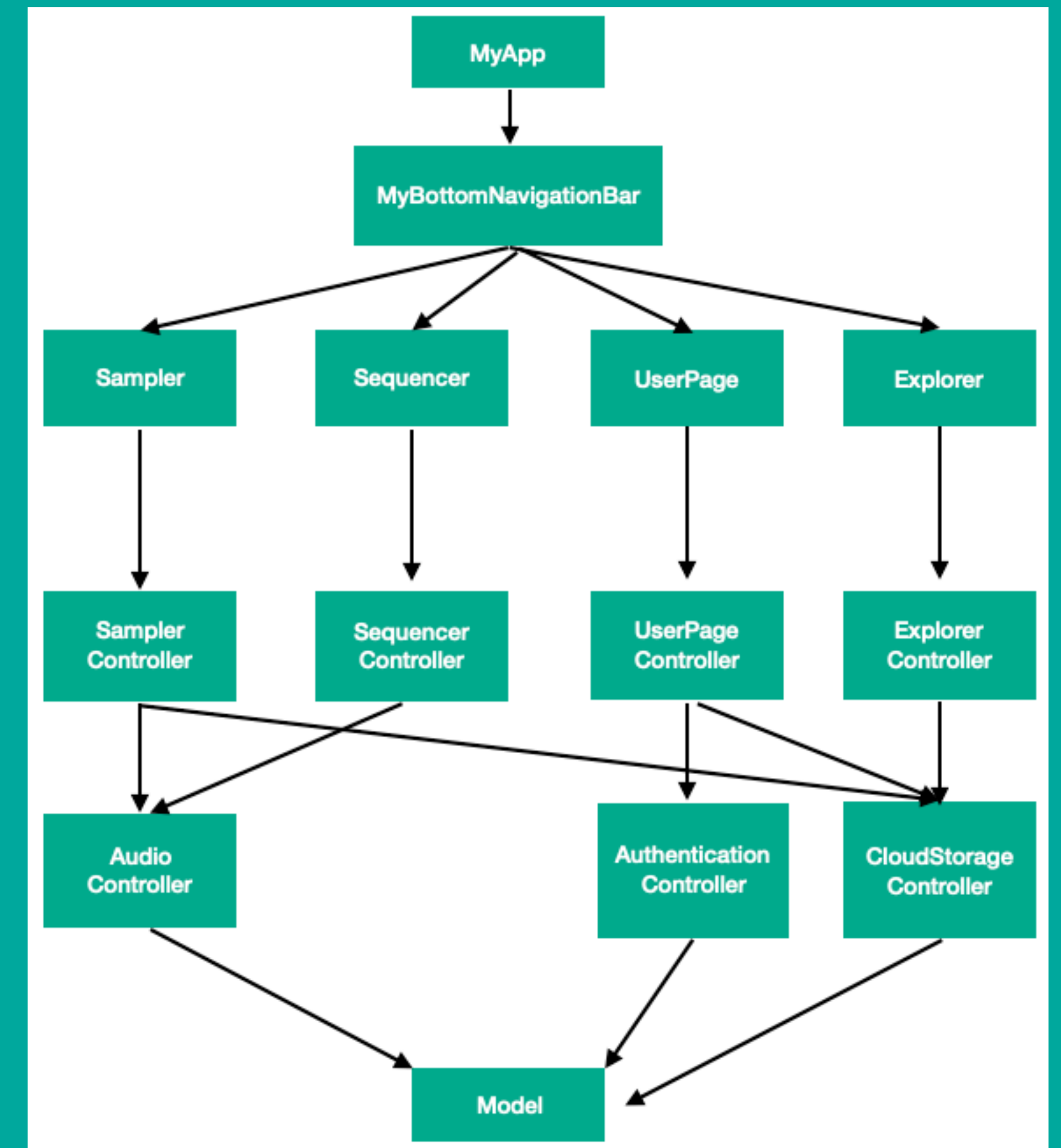
- Possibilità di registrare tracce audio (Samples) tramite il microfono del dispositivo e associarle automaticamente a un tasto dell'applicazione per la successiva riproduzione
- Rinominazione dei campioni
- Condivisione dei campioni con gli altri utenti iscritti alla piattaforma
- Upload su sistema di cloud (Google Drive per dispositivi Android, iCloud per iOS)
- Caricamento di file audio dal filesystem del dispositivo
- Sequencer
- Caricamento immagine del profilo tramite uso della fotocamera oppure selezione dal filesystem
- “Explorer” per la ricerca dei campioni condivisi, con possibilità di scaricarli, salvarli in un elenco di preferiti o riprodurli
- Sistema di filtraggio dei campioni presenti all'interno di Explorer
- Iscrizione alla piattaforma tramite servizi Google
- Iscrizione tramite indirizzo email con autenticazione dello stesso

# Struttura codice

Le classi di Simple Sample sono state progettate secondo il pattern architetturale “Model - View - Controller”, il quale permette di codificare classi con poco accoppiamento e quindi con compiti divisi.

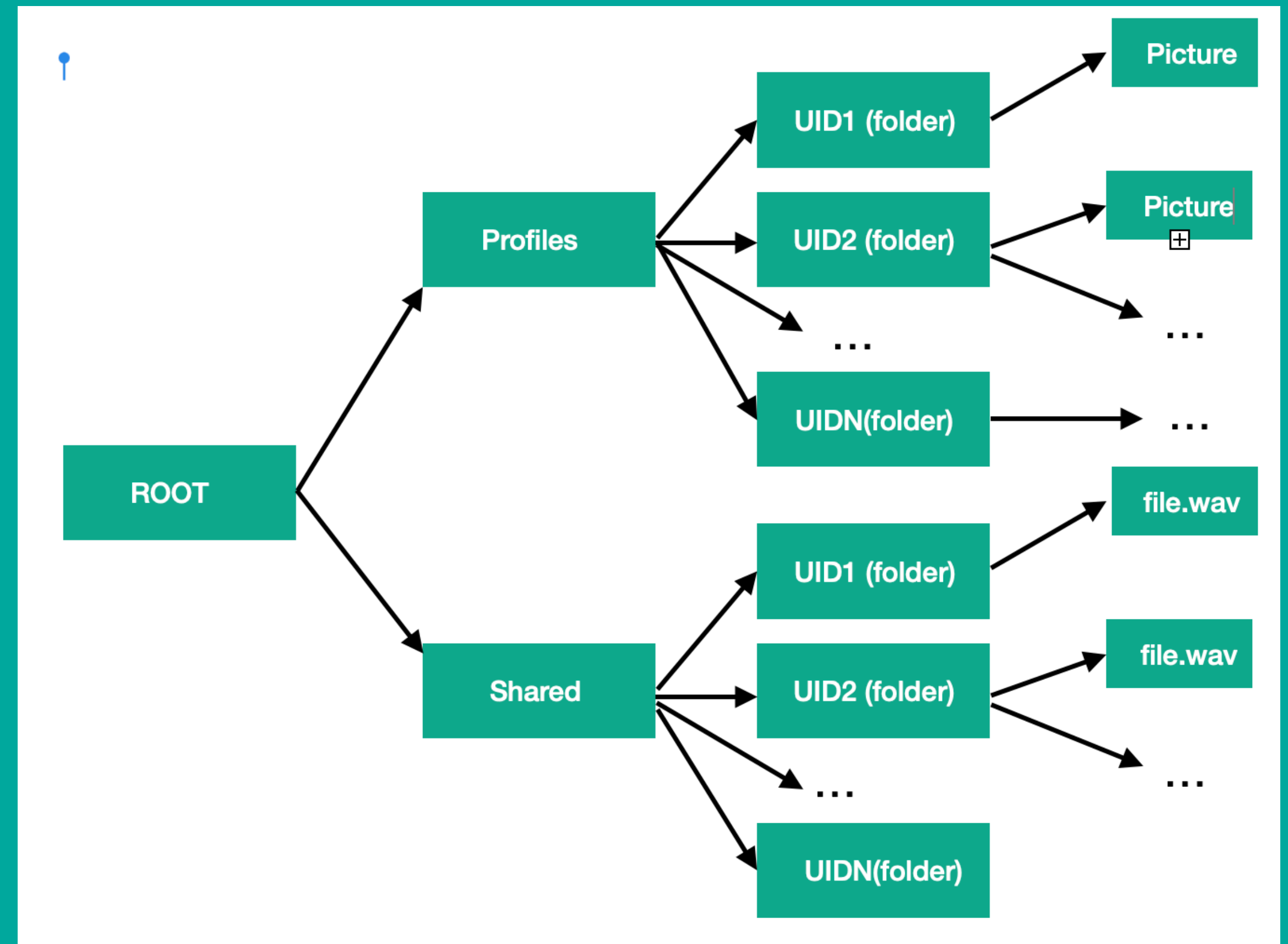
Le classi posso essere quindi divise nelle seguenti categorie:

1. Modelli: contengono i dati dell'applicazione
2. Controllers: governano la logica dell'applicazione facendo uso dei modelli. I controllers possono essere a loro volta divisi in:
  1. UI Controllers: gestiscono le operazioni attivate dall'utente tramite le interfacce grafiche
  2. Servizi: gestiscono le operazioni di background, come ad esempio il download dei dati da visualizzare o il login utente
3. Viste: classi per la visione dei dati. Rappresentano l'interfaccia con cui l'utente interagisce



# Salvataggio dei dati 1

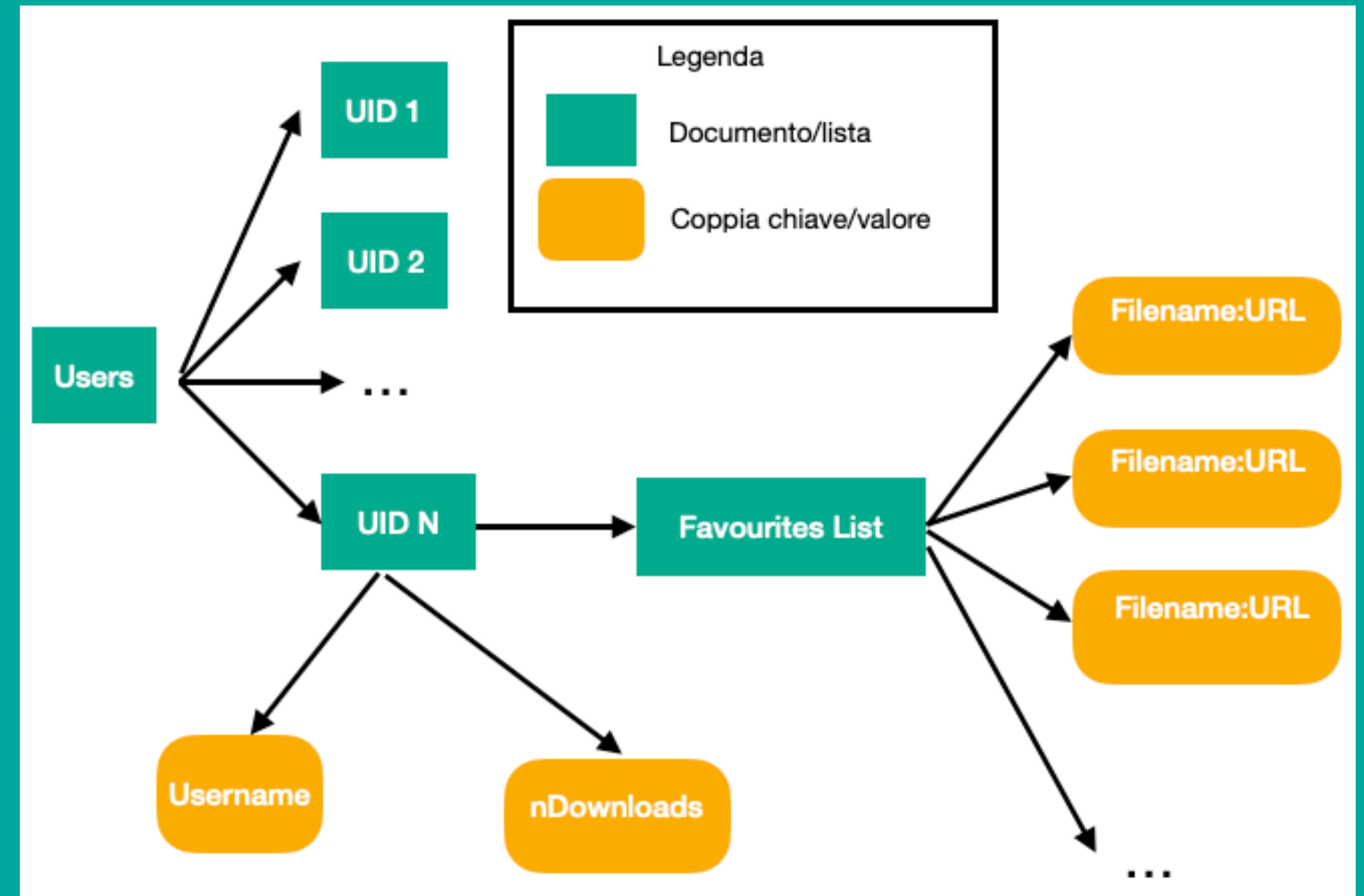
- Il salvataggio dei dati online fa uso dei servizi “Firebase”, integrati all’interno dell’applicazione tramite librerie apposite.
- Il salvataggio dipende dalla tipologia di dato
- File multimediali: informazioni come immagini del profilo o i sample condivisi sono salvate all’interno di “Firebase CloudStorage”, il quale si basa su una gerarchia ad albero
- 2 cartelle principali: “Shared” e “Profiles”. Ognuna contiene una cartella per ogni utente
- All’interno di “Shared” si mantengono i file condivisi, mentre in “Profiles” le immagini del profilo





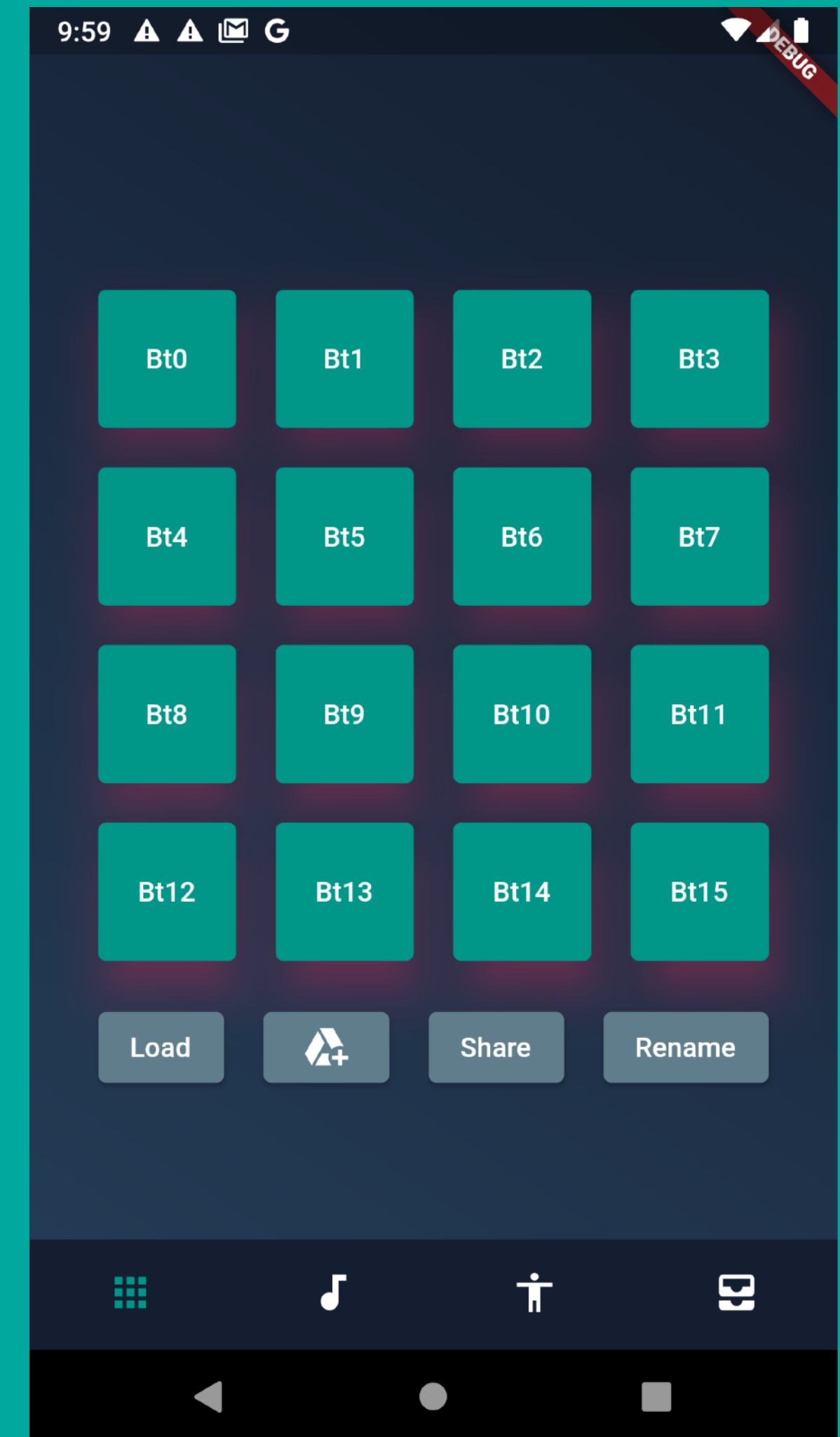
# Salvataggio dei dati 2

- Dati primitivi: salvati all'interno di "Firestore"
- Sistema di archiviazione "chiave - valore", il quale può contenere tipi primitivi o oggetti "documents"
- Ogni utente è dotato di un proprio documento nominato tramite il proprio identificativo univoco
- All'interno di ogni documento sono contenuti:
  - Nome utente
  - Numero totale di downloads dell'utente
  - Documento contenente la lista dei sample preferiti dall'utente selezionati tramite Explorer



# Sampler 1

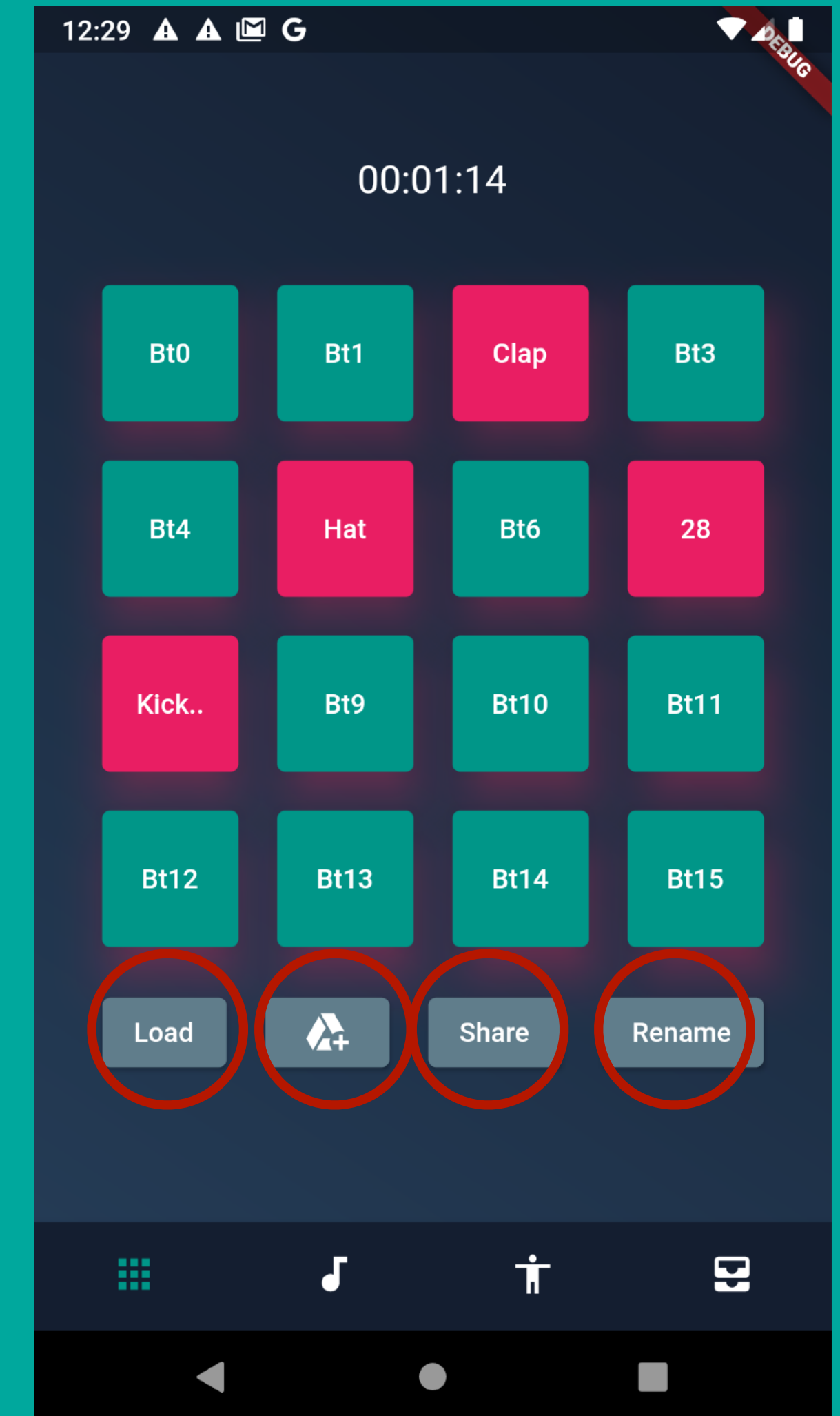
- Prima schermata visualizzata all'avvio dell'applicazione
- Consente di registrare samples tramite il microfono integrato o dispositivo audio collegato (cuffie con microfono) tenendo premuto un tasto della griglia. Tale sample creato è successivamente riproducibile premendo lo stesso tasto
- Consentita la riproduzione di più samples contemporaneamente premendo più tasti -> riproduzione dello stile di composizione MPC
- Una volta registrato, al file audio viene associato un nome secondo il pattern “*numero-registrazione.wav*”



# Sampler 2

4 tasti funzione:

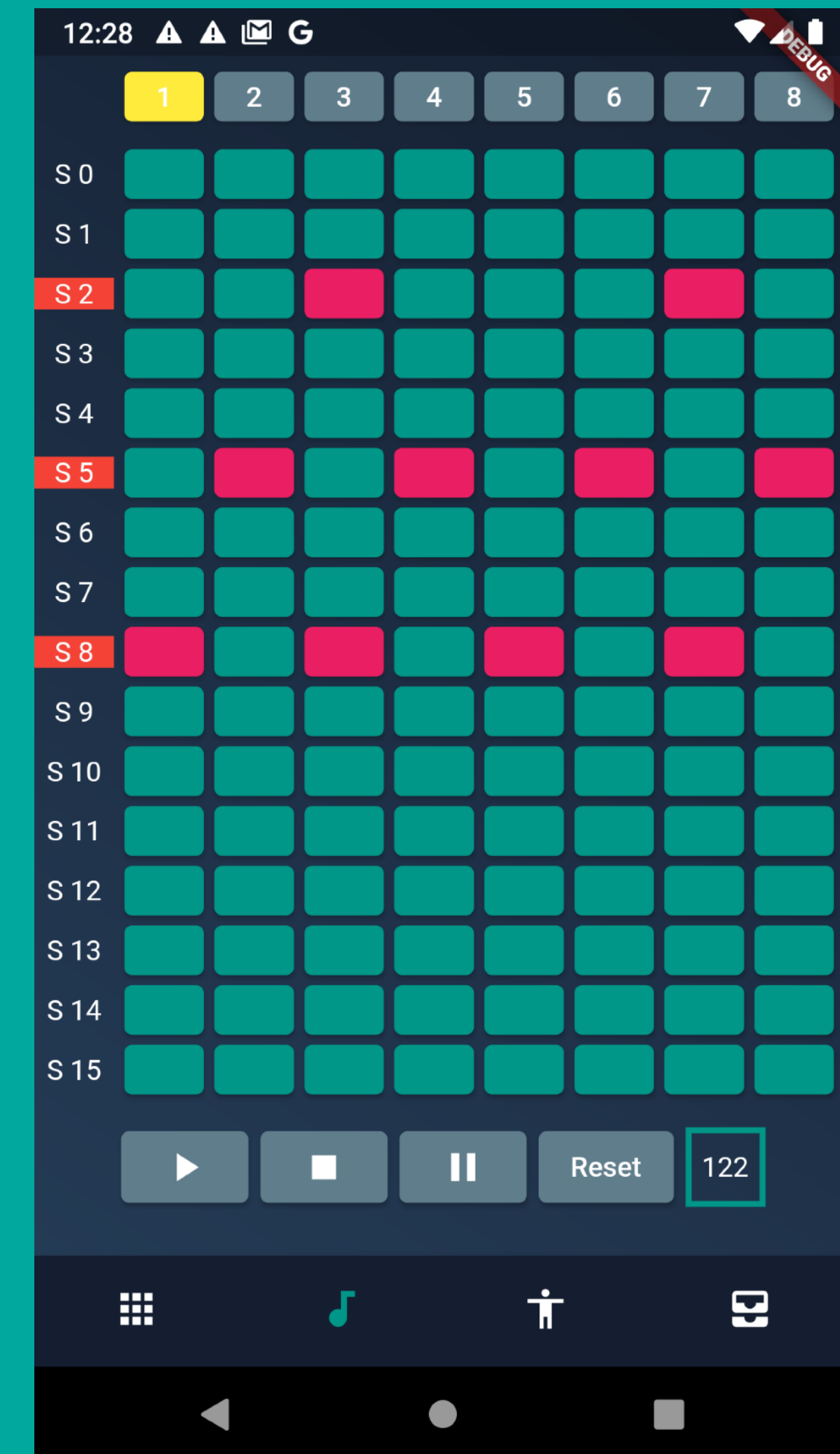
- Load: consente di eseguire 3 differenti azioni:
  1. Aprire un file picker per il caricamento di file audio (con estensione “wav”, “mp3”, “aac” e “m4a”) dal filesystem o dal servizio di cloud integrato
  2. Caricamento di assets audio integrati nell'applicazione: di default con Simple Sample viene fornito un set di campioni di una drum machine Roland 909
  3. Caricamento di file presenti nella cartella di salvataggio automatico dell'app
- Upload to Drive: caricamento nella cartella root di Google Drive
- Share: Condivisione di un campione, a cui sono forniti un nuovo nome scelto dall'utente e una lista di tag. Una volta caricato, il sample è accessibile tramite l'Explorer a tutti gli utenti iscritti
- Rename: associazione di un nuovo nome al campione collegato a un tasto





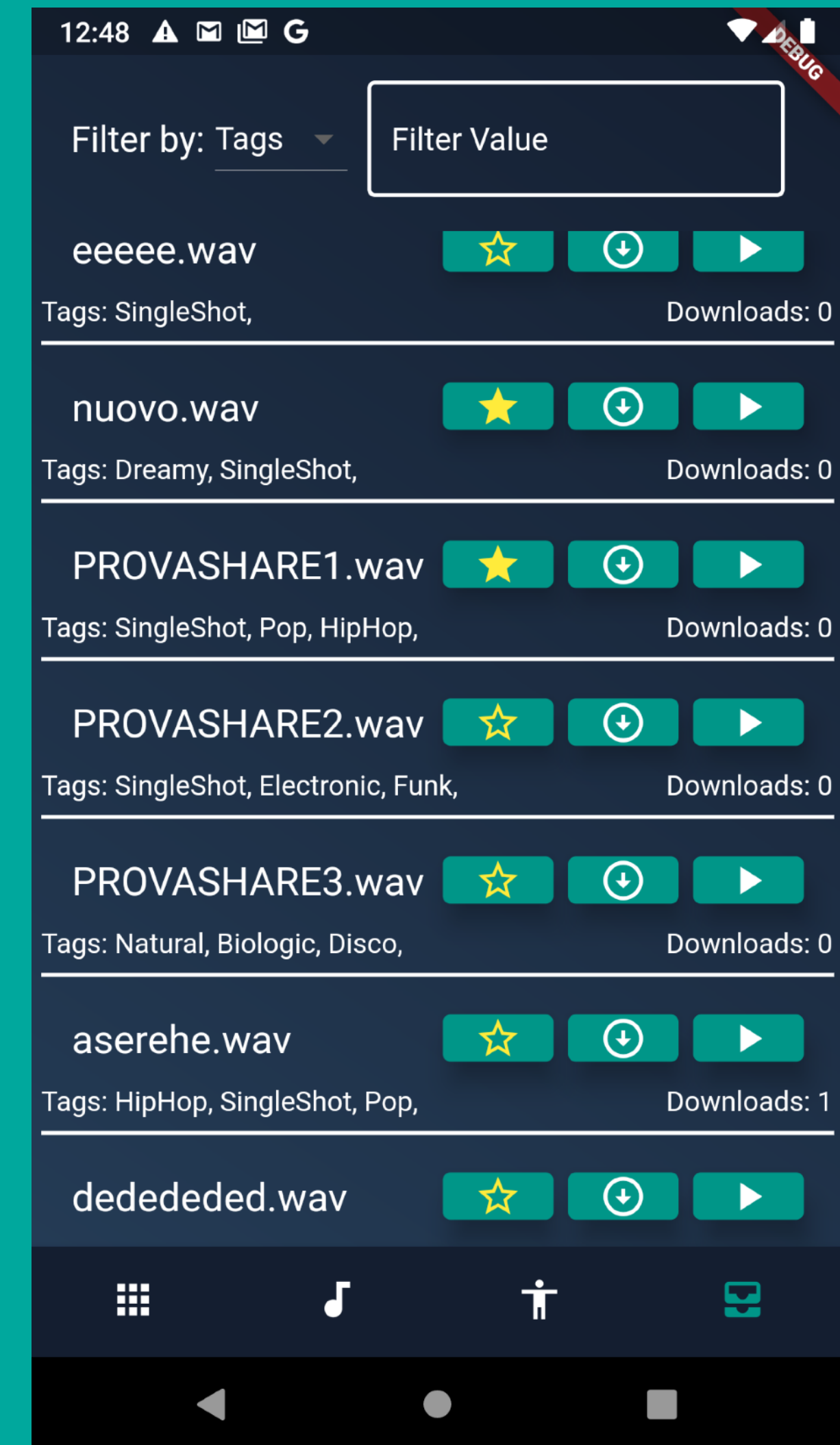
# Sequencer

- Griglia di tasti per la riproduzione automatica dei suoni assegnati ai tasti del Sampler
- Banco di prova per i campioni registrati per vedere come si comportano all'interno di una ritmica (ricordo che lo scopo di Simple Sample è di “raccoltore” di suoni da riportare all'interno di uno studio di registrazione)
- Ispirato al funzionamento di un reale sequencer
- Griglia di tasti composta da 16 righe (una per bottone del Sampler) e 8 colonne
- 4 tasti funzione: “play”, “pausa”, “stop” e “reset” (ripulisce la griglia)
- Possibilità di definire i BPM (numero di battiti per minuto) a cui il sequencer deve riprodurre i campioni
- Funzionamento: una volta premuto “play”, il sequencer riproduce una colonna per volta in modo ciclico. Per ogni colonna, riproduce tutti i campioni il cui tasto risulta essere attivato (colore viola)



# Explorer

- Schermata per la visualizzazione dei campioni condivisi da tutti gli utenti iscritti
- Lista di elementi in cui ognuno è strutturato mostrando:
  - Nome del campione
  - Lista di tag associati
  - Numero di volte in cui è stato scaricato
  - 3 tasti funzione:
    1. Add to Favorites: Aggiunge il campione tra i “preferiti” dell’utente con conseguente visualizzazione all’interno della propria pagina utente
    2. Download: scarica il campione all’interno della cartella di default di Simple Sample
    3. Play: riproduzione del campione
- Sistema di filtraggio attraverso tag oppure nome dei campioni

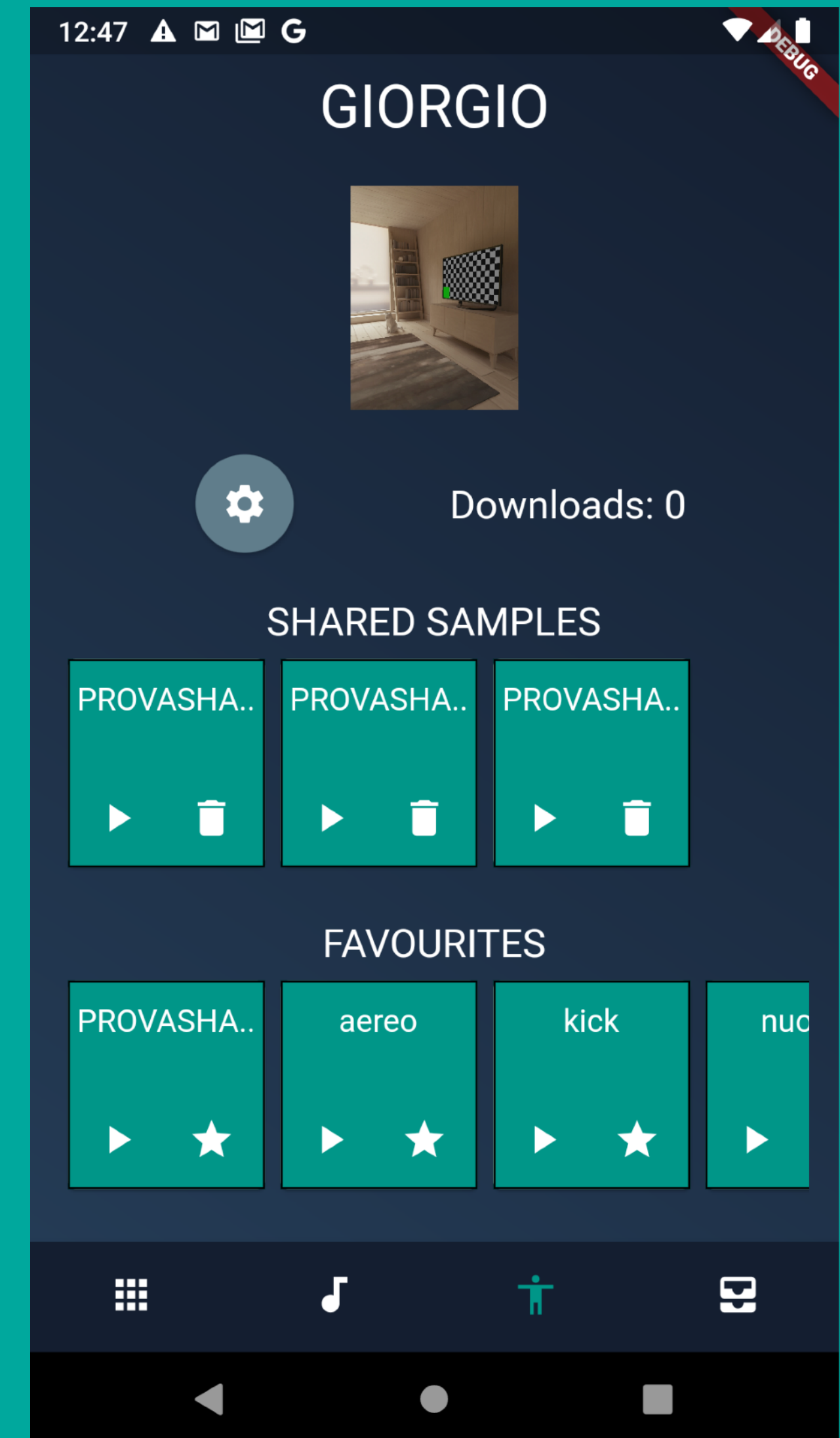


# User Page

Pagina contenente tutte le informazioni dell'utente:

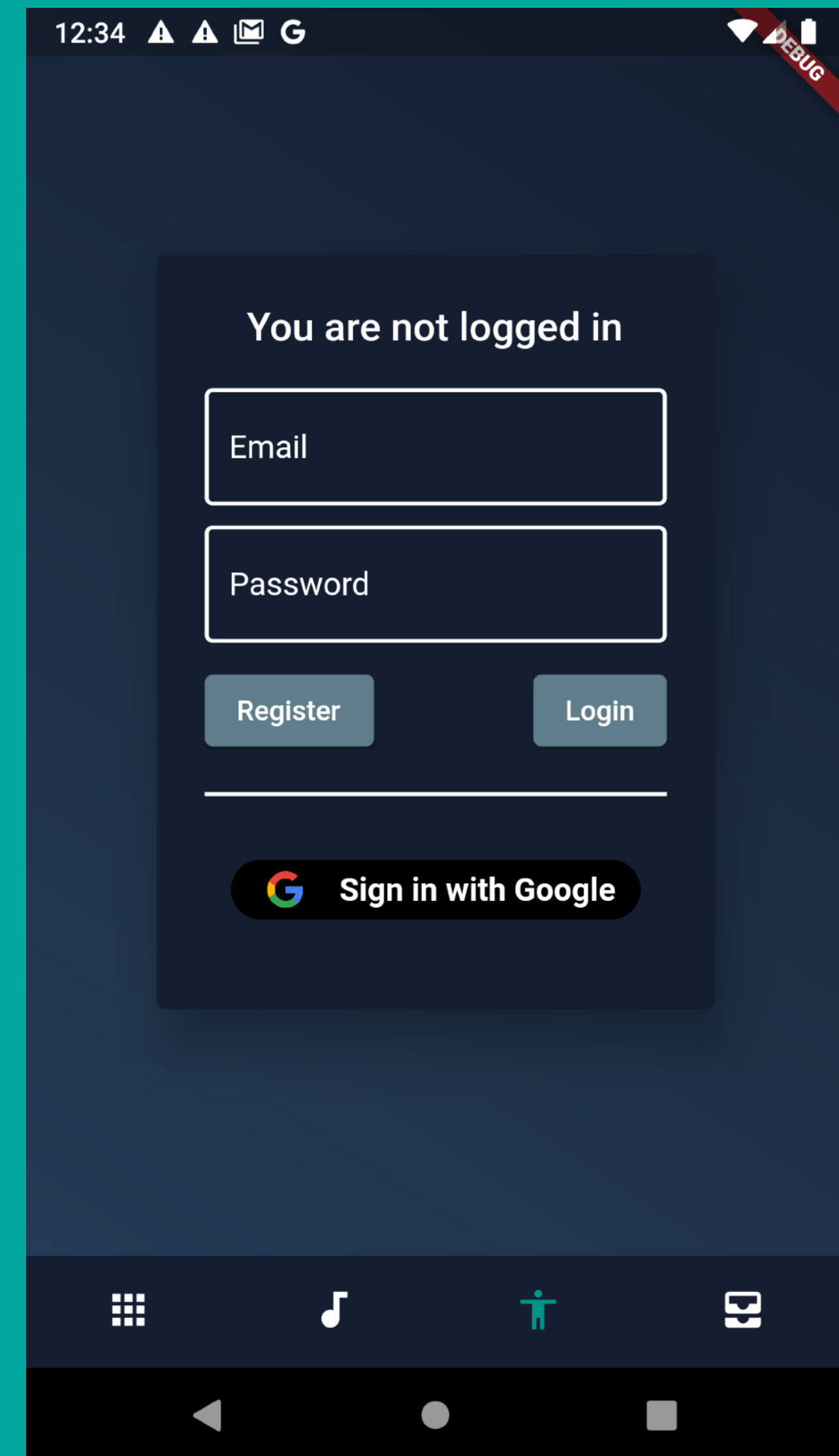
- Nome utente (modificabile tramite il menu “Impostazioni”)
- Immagine del profilo
- Numero di downloads: numero totale di volte in cui i campioni condivisi dall'utente sono stati scaricati
- Shared samples: lista dei campioni condivisi. Per ogni campione viene visualizzato il nome, un tasto play per la riproduzione e un bottone per la rimozione: esso rimuove il campione anche dal server, non rendendolo più disponibile agli altri utenti
- Favourites: lista dei campioni preferiti. Per ognuno viene visualizzato il nome, un tasto per la riproduzione e uno per la rimozione

Dalla User Page cliccando sul tasto circolare è possibile accedere a un menu “Impostazioni” in cui, tra le altre cose, è possibile cambiare nome utente, cambiare immagine del profilo, o eseguire il logout



# Login

- Schermata visualizzata quando l'utente non è collegato oppure ha effettuato il logout
- Possibile l'accesso alla piattaforma tramite Email e Password oppure tramite account Google
- Se scollegato, l'utente non può eseguire la condivisione di campioni o accedere all'Explorer
- Se l'utente non è iscritto alla piattaforma tramite account Google, non può effettuare il caricamento su Google Drive





# Testing

L'applicazione è stata sviluppata secondo un approccio Test Development Driven (TDD), basato sul test dei metodi una volta completata la loro implementazione. Le tipologie di test effettuati sono:

1. Unit Test: esame di singoli test non banali
2. Widget Test: verifica dei Widget che compongono l'interfaccia grafica
3. Integration Test: test che include l'azione di più metodi che collaborano tra di loro al fine di eseguire una "macroiterazione", che richiede la collaborazione di più parti al fine di raggiungere un obiettivo comune
4. End-to-End Test: verifica che simula l'interazione dell'utente con il sistema