ACEA TIMBRATURE (versione mockata)

Panoramica sull'applicazione

Puntamenti

L'applicazione punta a diversi sistemi utilizzando diversi flavours di build, configurati nel *build.gradle* sotto la cartella app.

Android Activities

L'applicazione è estremamente semplice e si compone delle seguenti activity

- LoginActivity.java
- MainActivity.java
- SplashScreen.java
- TimbraturaActivity.java

Splash Screen

Prima activity che viene invocata all'apertura dell'app. Verifica se esiste un verification code: in caso affermativo apre la Main Activity, altrimenti lancia la Login Activity.

Login Activity

LoginActivity.java offre le funzionalità di accesso ed autenticazione dell'utente:

- permette all'utente l'inserimento della matricola per iniziare la registrazione;
- può essere aperta attraverso un Intent per completare il processo di registrazione.

L'apertura con Intent viene invocata con il tap sul link di verifica ricevuto dall'utente tramite email.

Main Activity

Rappresenta la schermata principale dell'app:

- permette l'apertura dell'activity per effettuare una nuova timbratura;
- Consente di visualizzare un calendario nonché le timbrature del giorno selezionato.

Timbratura Activity

Come suggerisce il nome, consente all'utente di effettuare una timbratura ed inviarla al sistema.

HTTP Client

Il client HTTP utilizzato è **Ion**. Le chiamate di rete sono gestite all'interno della classe NetworkManager.java.

Descrizione dell'applicazione

TimbratureService

- Interfaccia per i metodi GET (**getTimbrature()** e **getVerificationCode()**) e POST (**sendTimbratura(**timbJson**)**)

TimbratureNetworkModel

- Classe per la gestione delle richieste/risposte HTTP.

PrototypingInterceptor

- Classe che si occupa dell'intercettazione delle richieste HTTP e della costruzione delle risposte HTTP;
- Genera la lista di timbrature a partire dai dati in formato JSON salvati negli "assets" e dalle eventuali timbrature inviate tramite l'applicazione dall'utente (memorizzate nelle SharedPreferences);
- Controlla la matricola inserita dall'utente al momento del login e restituisce il verification code (nonché il token, che –però- non viene utilizzato).

LoginActivity

- Consente all'utente di effettuare il login "bypassando" la richiesta del verification code e, successivamente, dell'authentication token ad esso associato;
- la richiesta del verification code avviene dopo l'inserimento della matricola specificata dall'app stessa ("ABC123") ed è realizzata attraverso il metodo **getVerificationCode()** dell'oggetto **TimbratureNetworkModel**;
- la classe **TimbratureNetworkModel** offre le funzionalità per la richiesta e l'ottenimento del codice di verifica, nonché per l'acquisizione delle timbrature presenti a sistema (memorizzate, in tale versione mock, nelle SharedPreferences) e per l'invio di una nuova timbratura (anch'essa memorizzata nelle SharedPreferences);
- una volta terminata la fase di login, viene lanciata l'activity principale dell'applicazione.

MainActivity

- Mostra le timbrature -ottenute tramite gli oggetti descritti in precedenza- sul calendario;
- Consente la creazione di una timbratura (memorizzata poi nelle SharedPreferences): al momento della creazione viene lanciata l'activity TimbraturaActivity.

TimbraturaActivity

- Crea l'oggetto JSON "Timbratura" e lo invia invocando il metodo **sendTimbratura()** affinchè sia memorizzato nelle SharedPreferences attraverso gli oggetti descritti in precedenza.