**ANALISI DI FATTIBILITA’**

Di seguito i punti analizzati:

* **Componenti utilizzati per l’interfaccia grafica:**

Per l’interfaccia grafica abbiamo deciso di utilizzare come Layout principale il ConstraintLayout perché, avendolo utilizzato spesso, ci sentiamo più a nostro agio con questo tipo di Layout, inoltre è molto più semplice e flessibile rispetto al RelativeLayout.

Per quanto riguarada la lista dei film abbiamo deciso di utilizzare una ListView in modo da rafforzare l’argomento in vista dell’esame finale ed anche perché non vogliamo correre il rischio di perdere molto tempo nell’implementazione della RecyclerView che conosciamo poco.

* **Quale servizio API utilizzare e perché:**

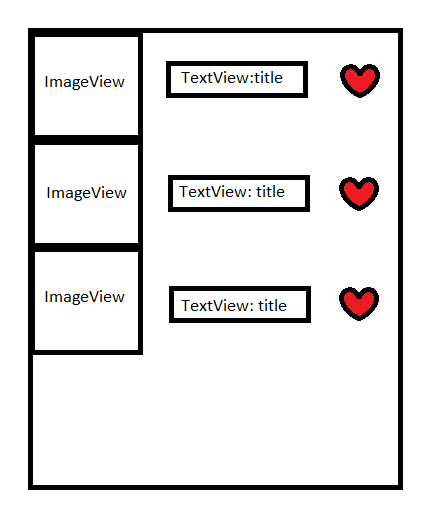
Abbiamo deciso di utilizzare il servizio The Movie Database perché ha una documentazione molto più completa rispetto a OMDb. Inoltre ha la possibilità di reperire i dati per page che ci ritornerebbe utile nel caso implementassimo un meccanismo endless scroll. Altri vantaggi di the movie Db sono che non ha un limite di chiamate ed ha una sezione dedicata al supporto e una community che la segue, mentre OMDb concede solo 1000 chaimate giornaliere free e per il supporto ha solo la possibilità di contattare l’unica persona che ha creato e segue il Db.

* **Quale libreria per le immagini utilizzare e perché:**

Abbiamo deciso di utilizzare la libreria glide perché ha delle performance molto migliori rispetto a Picasso ed inoltre si adatta molto meglio all’utilizzo delle listView minimizzando il rischio di andare in OutOfMemory. Glide inoltre è molto più personalizzabile ed ha molte funzionalità aggiuntive che per questo progetto non sono necessarie, ma che in futuro potrebbero tornare utili e quindi preferiamo utilizzare Glide per imparare ed iniziare a conoscere questa libreria. Per quanto riguarda la documentazione, entrambe sono chiare ma quella di Glide è molto più vasta.

* **Funzionalità extra:**

Abbiamo deciso di implementare la possibilità di aggiungere un film ai preferiti, aggiungendo una tabella preferiti nel database. Con il long tap su un elemento della listView aggiungeremo l’elemento ai preferiti. Con il click su un bottone in basso a destra (floatingButton) apriremo un dialog Fragment con l’elenco di tutti i preferiti(vedi immagine sotto).



Premendo sul cuoricino questo diventerà riempito di bianco e eliminerà il campo dai preferiti.

* **Memorizzazione dei dati in locale:**

Abbiamo deciso di utilizzare il ContentProvider per rafforzare la nostra conoscenza dell’argomento in vista dell’esame. Inoltre cosente di utilizzare il LoaderManager ed altre funzionalità utili per la gestione dei dati.

* **Chiamate http:**

Abbiamo deciso di utillizzare Retrofit per le chiamate http, perché è l’unica tecnologia che al momento conosciamo ed è supportata da google, quindi molto affidabile. Inoltre mostra un grande potenziale per le implementazioni.

* **Gestione della memorizzazione delle immagini:**

Abbiamo deciso di salvare tutti i dati che l’utente scaricherà nel corso dell’uso dell’app in modo da limitare il numero di dati scaricati da internet. Tutto ciò a lungo andare però appesentirebbe l’applicazione. Per ovviare a questo problema abbiamo deciso che quando l’utente chiude l’applicazione manteniamo solo i film preferiti e le prime 5 page e i restanti dati vengono eliminati e ricaricati solo in caso l’utente a una nuova riapertura dell’app ne abbia bisogno.