ANDROID FINAL PROJECT – JOKE GENERATOR  
Bhavna Achar | 103993200 | bachar@myseneca.ca

The app uses the [JokeAPI](https://sv443.net/jokeapi/v2/) to generate random jokes, which can be filtered based on either certain categories or words that the joke must contain. Once generated, the user is able to save their favorite jokes.  
  
The app consists of the following components:

* 1 activity for the main joke generator
* 1 activity for the saved jokes view
* 3 fragments for the category selection, for the search string, and for displaying the selected joke and category from the saved jokes view
* Recycler View for displaying the saved jokes list, along with an adapter
* A Joke class which is the entity for the DB (implements Parcelable)
* A database manager class
* A Database abstract class which defines the entity of the DB and connects with Room DB
* A DAO interface
* A networking manager class that sets up the connection and makes the API call and retrieves the response as a string
* A JSON manager which parses the string retrieved from the networking manager class and creates the joke object

Below are some of the main functionalities of the app:

* **Filtering by Category**: the user can select one or multiple categories, and the generator will provide jokes from any one of the categories. The categories are displayed on the screen using a DialogFragment, which contains a callback interface to send back the selected categories to the MainActivity.  
    
  Example URL: https://v2.jokeapi.dev/joke/Programming,Miscellaneous,Pun
* **Filtering by Word:** the user can enter a word/phrase and the generator will retrieve only those jokes that contain the entered string. The text box for entering the string is displayed with a DialogFragment, which contains a callback interface to send the selected string back to MainActivity.  
    
  Example URL: <https://v2.jokeapi.dev/joke/Any?contains=dog>
* **Error Handling:** if the API returns an error (in the case of a joke with the given filters not existing) the app instead displays an error text which is in red, and the save button is disabled so that an invalid joke is not saved to the database. This is done using the “error” key of the response in an if-else clause in the JsonManager class.  
    
  Example API Response for an error:  
    
  {  
   "error": true,  
   "internalError": false,  
   "code": 106,  
   "message": "No matching joke found",  
   "causedBy": [  
   "No jokes were found that match your provided filter(s)."  
   ],  
   "additionalInfo": "Error while finalizing joke filtering: No jokes were found that match your provided filter(s).",  
   "timestamp": 1682129304524  
  }
* **Save:** if the user comes across a joke that they like, they can save it locally on the app. Persistent storage is made possible using Room DB.
* **Switch between Joke Generator and Saved Jokes**: from the menu, the user can move between the two screens. This was done by creating a menu resource folder with a menu file that contains the two menu options, each having a click listener to open the respective activities via an intent.
* **Swipe to Delete:** the user can swipe on a saved joke to delete it from the app
* **Spanish Language Support**: the app supports Spanish – apart from adding a values-es directory with a strings.xml file for the apps’ strings’ translations, the API itself supports multiple languages. By passing the parameter ?lang=”es”, the API retrieves only Spanish jokes.  
    
  Example URL: https://v2.jokeapi.dev/joke/Any?lang=es