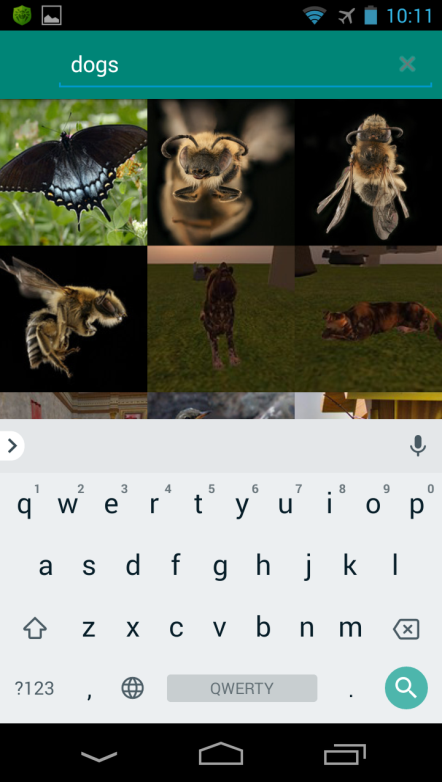
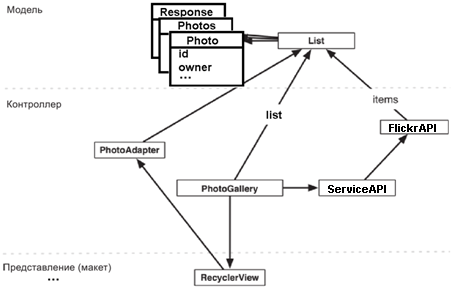
**Лабораторная работа**. Создать приложение **PhotoGallery**

1. Создать приложение с шаблоном **EmptyActivity.**
2. Выполните рефакторинг и замените название MainActivity на **PhotoGallery**, а макет activity\_main соответственно на **gallery\_activity**.
3. Зарегистрируйтесь на сайте flickr.com и получите ключ (api\_key). Ознакомьтесь с [документацией](https://www.flickr.com/services/api/) по архитектуре Web-сервиса Flickr.
4. Откройте flickr.photos.getRecent [API Explorer](https://www.flickr.com/services/api/explore/flickr.photos.getRecent) и получите JSON.
5. С помощью следующего [Web-сервиса](http://www.jsonschema2pojo.org/) сформируйте классы объектов для парсинга JSON и добавьте эти классы в пакет **model** вашего проекта, убрав аннотации.



1. Добавьте в код PhotoGallery объект **RecyclerView** для списка фотографий и создайте класс **PhotoAdapter** для реализации RecyclerView. В качестве LayoutManager для RecyclerView установите GridLayoutManager (таблица с 3 столбцами).
2. Создайте папку api в вашем проекте, в ней создайте интерфейс FlickrAPI и класс ServiceAPI:

public **interface** FlickrAPI {

@GET("services/rest/?method=flickr.photos.getRecent&api\_key=Your API

&extras=url\_s&format=json&nojsoncallback=1")  
 Call<PhotosResponse> getRecent()**;**@GET("services/rest/?method=flickr.photos.search&api\_key=d Your API

&extras=url\_s&format=json&nojsoncallback=1")  
 Call<PhotosResponse> getSearchPhotos(@Query("text") String keyWord)**;**}

public **class** ServiceAPI {  
 private static final String *BASE\_URL* = "https://api.flickr.com/"**;** private static Retrofit *retrofit* = null**;** public static Retrofit getRetrofit() {  
 if (*retrofit* == null) {  
 Gson gson = new GsonBuilder()  
 .setLenient()  
 .create()**;** *retrofit* = new Retrofit.Builder()  
 .baseUrl(*BASE\_URL*)  
 .addConverterFactory(GsonConverterFactory.*create*(gson))  
 .build()**;** return *retrofit***;** }  
 return *retrofit***;** }  
}

1. Добавьте в класс Photo аннотации **@Entity** и определите первичный ключ с помощью аннотации **@PrimaryKey**.
2. Добавьте папку db и переопределите в ней интерфейс DAO и абстрактный класс для вашей БД.

@Dao  
public **interface** PhotosDao {  
 @Query("SELECT \* FROM Photo")  
 public List<Photo> LoadAll()**;** @Insert(onConflict = OnConflictStrategy.*REPLACE*)  
 public void insertPhoto(Photo photo)**;** @Delete  
 public void deletePhoto(Photo photo)**;**}

@Database(entities = {Photo.class}**,** version = **1,** exportSchema = false)  
public **abstract class** PhotosDB extends RoomDatabase {  
 public abstract PhotosDao photoDao()**;** private static volatile PhotosDB *INSTANCE***;** private static final int *NUMBER\_OF\_THREADS* = **4;** static final ExecutorService *dbWriteExecutor* =  
 Executors.*newFixedThreadPool*(*NUMBER\_OF\_THREADS*)**;** static PhotosDB getDatabase(final Context context) {  
 if (*INSTANCE* == null) {  
 synchronized (PhotosDB.class) {  
 if (*INSTANCE* == null) {  
 *INSTANCE* = Room.*databaseBuilder*(context.getApplicationContext()**,** PhotosDB.class**,** "database")  
 .build()**;** }  
 }  
 }  
 return *INSTANCE***;** }  
}

1. Добавьте в приложение возможность поиска фотографий на Flickr. Добавьте выпадающее меню на панель инструментов с элементом SearchView для поиска.
2. Добавьте в адаптер RecyclerView слушатель для выбора элемента из списка и переопределите в нём метод, который будет добавлять выбранный элемент в локальную БД.
3. Добавьте в меню пункт, который будет выдавать в список RecyclerView фотографии из локальной БД.