НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

з дисципліни

“Програмування мобільних систем”

Виконав:

студент групи ІВ-83

ЗК ІО-7303

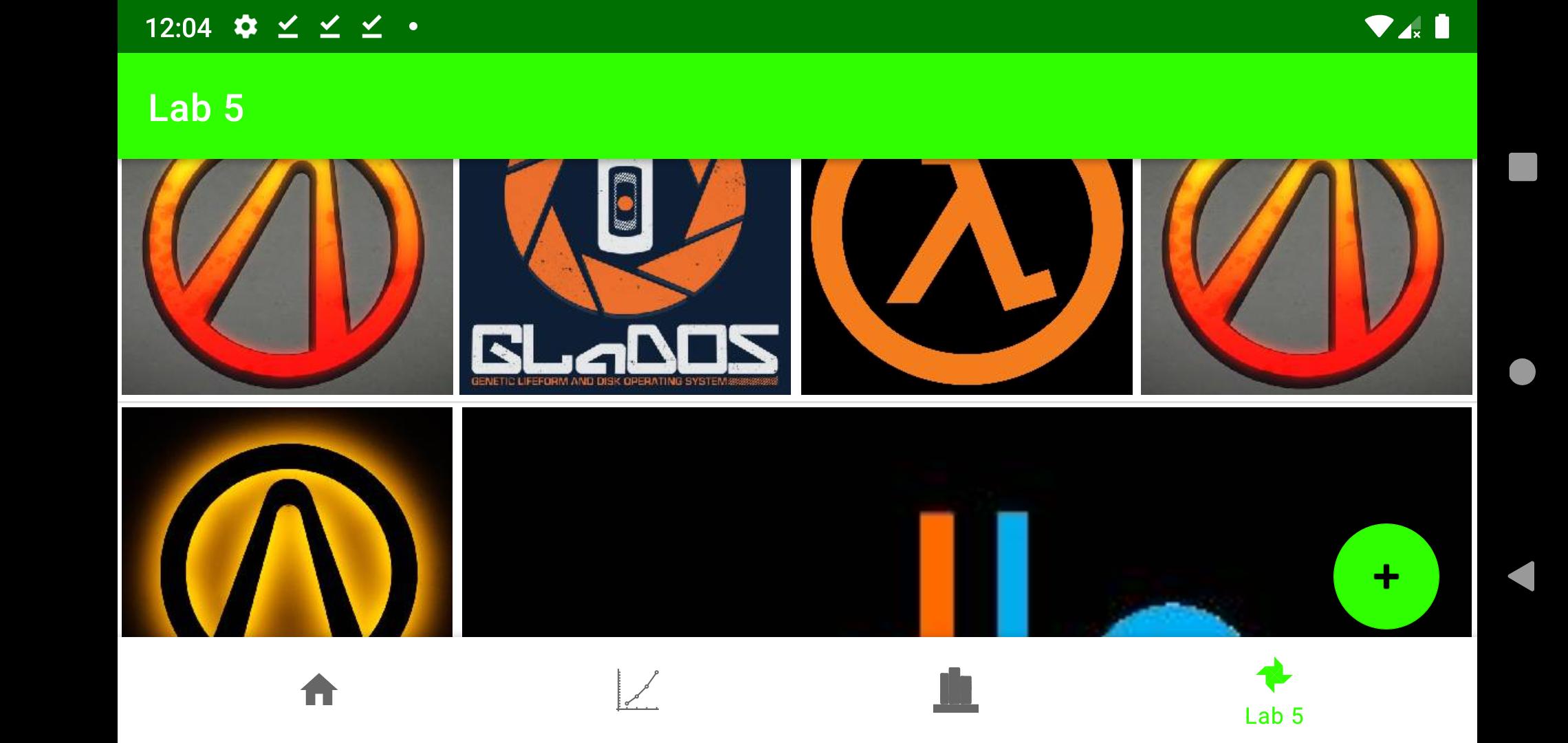
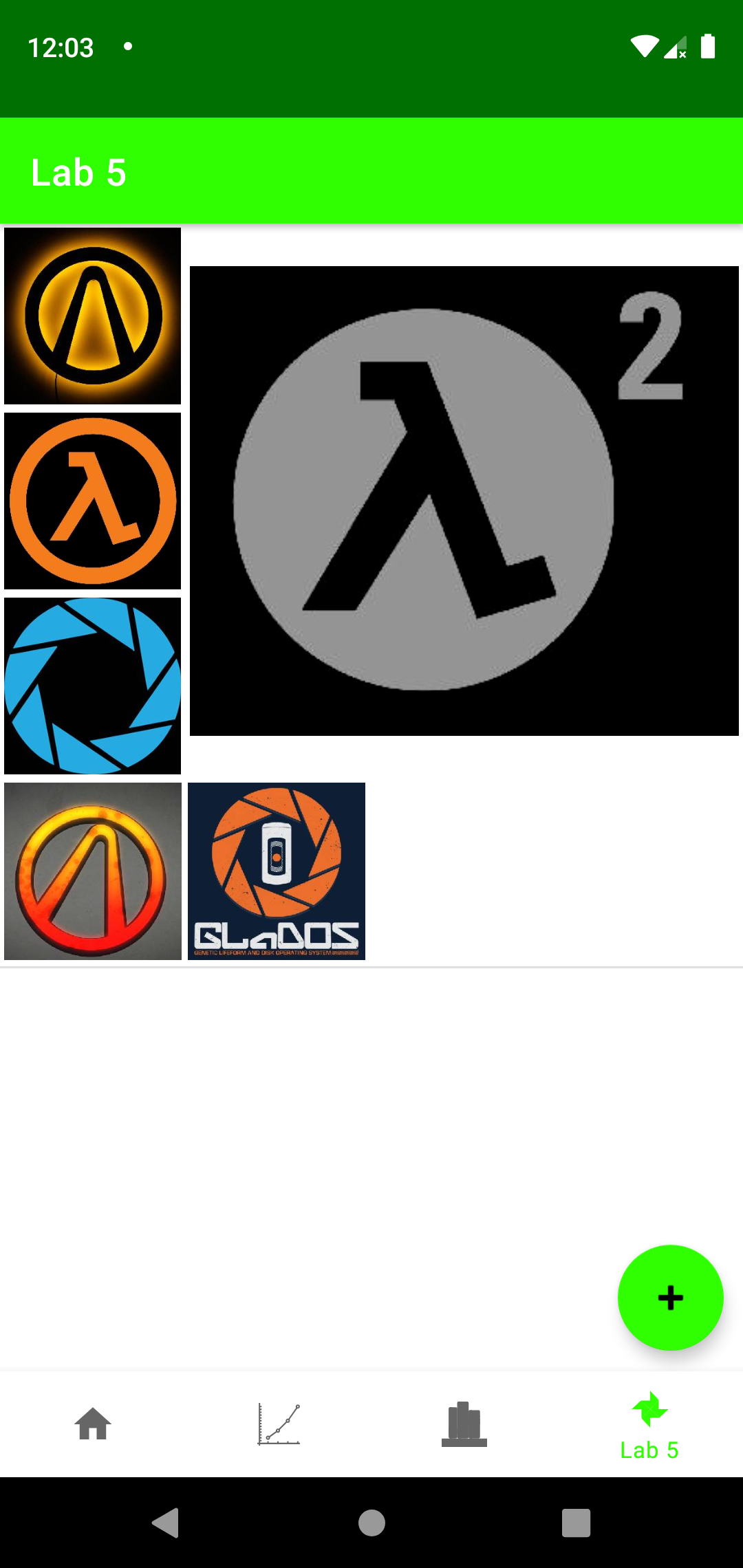
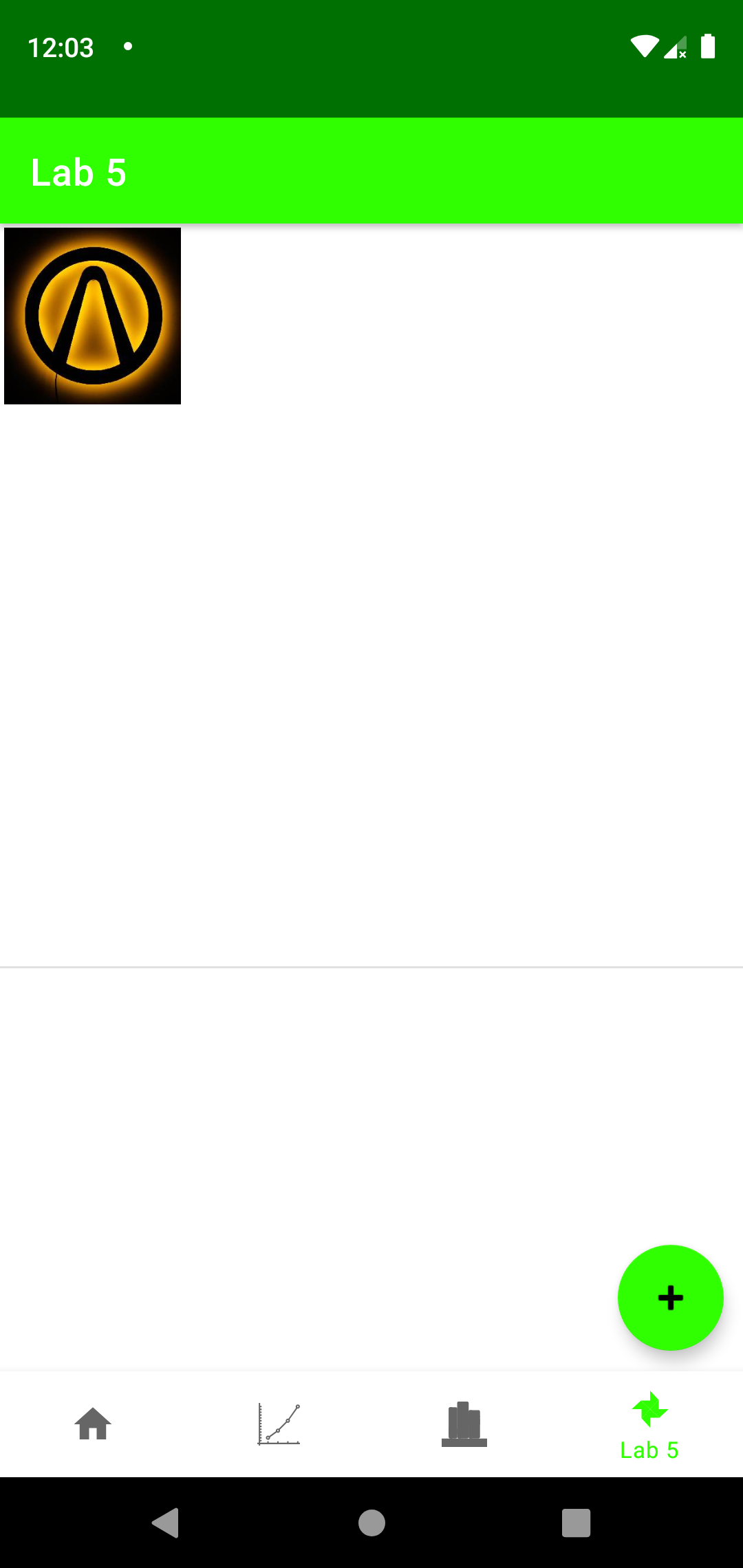
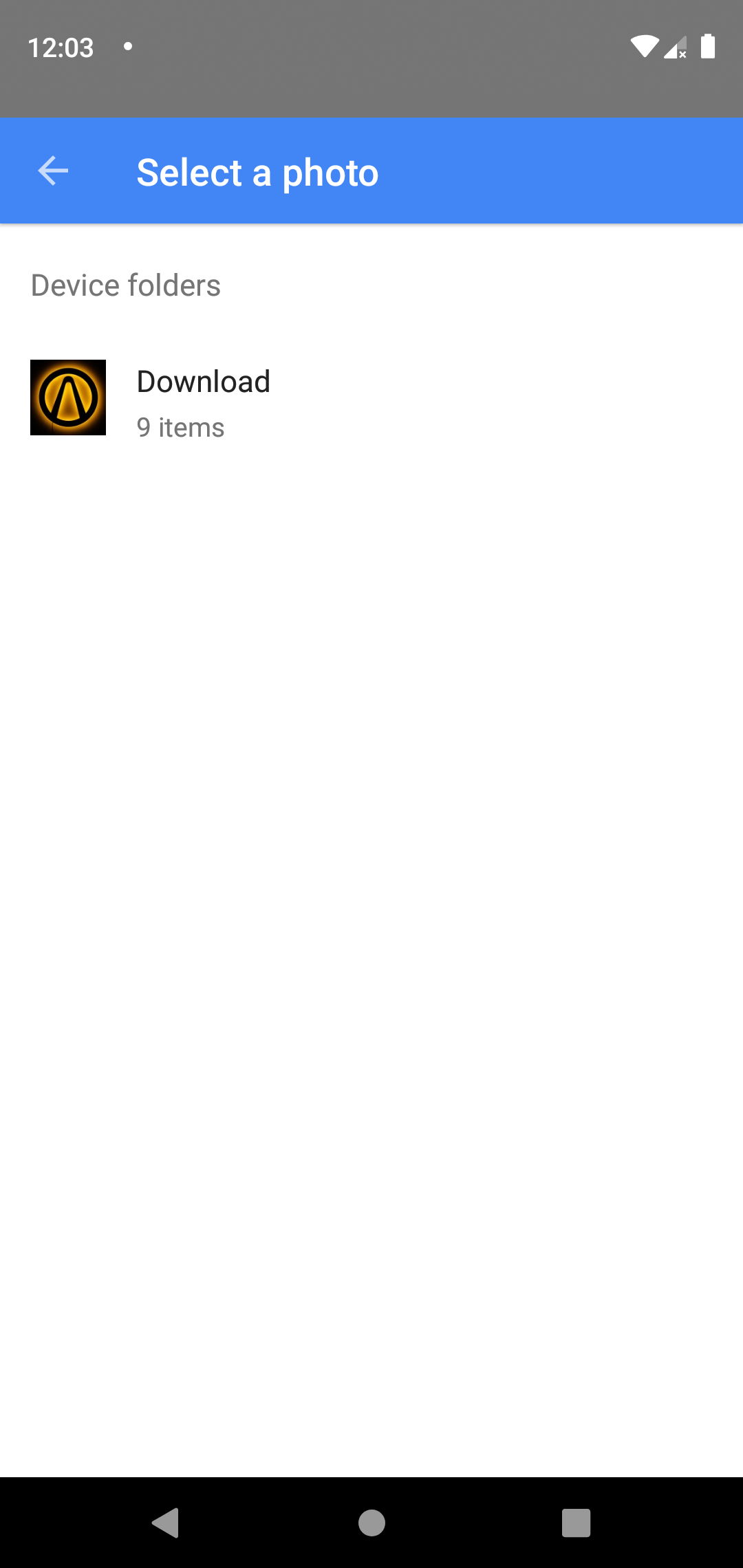
Ву В'єт Тунг

Київ 2021

**Варіант № 7303%6+1=2**

|  |
| --- |
| **Варіант 2** |
|  |

**Скріншоти роботи додатка**



**Лістинг коду**

**ImagesFragment.java**

package ua.kpi.comsys.IO7303.ui.gallery;  
  
import android.annotation.SuppressLint;  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Point;  
import android.net.Uri;  
import android.content.SharedPreferences;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.Bundle;  
import android.view.Display;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
  
import androidx.fragment.app.Fragment;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
  
import ua.kpi.comsys.IO7303.R;  
  
import static android.app.Activity.*RESULT\_OK*;  
  
public class GalleryFragment extends Fragment {  
 View root;  
 static int *width*;  
 int height;  
 List<List<String>> workImagesList = new ArrayList<>();  
 private final int success = 1;  
 private ImagesListAdapter adapter;  
 String elemsSettingsName = "collections";  
 ListView listView;  
 Boolean window\_active = false;  
 static LinearLayout *layout*;  
  
 public String imagesListsToString(List<List<String>> img){  
 StringBuilder result = new StringBuilder();  
 for (List<String> obj1 : img)  
 for (int i = 0; i < obj1.size(); i++) {  
 result.append(obj1.get(i));  
 result.append(";");  
 }  
 String resultStr = result.toString();  
 if (resultStr.length()>0)  
 return resultStr.substring(0, resultStr.length() - 1);  
 return "";  
 }  
  
 public List<List<String>> getImagesListsFromString(String imgStr){  
 List<List<String>> result = new ArrayList<>();  
 List<String> firstStep = new ArrayList<String>(Arrays.*asList*(imgStr.split(";")));  
  
 if (imgStr.equals(""))  
 return result;  
  
 for (String obj : firstStep) {  
 if (result.size() == 0) {  
 List<String> tempImageList = new ArrayList<>();  
 result.add(tempImageList);  
 }  
 if (result.get(result.size() - 1).size() >= 8) {  
 List<String> tempImageList = new ArrayList<>();  
 tempImageList.add(obj);  
 result.add(tempImageList);  
 } else {  
 result.get(result.size() - 1).add(obj);  
 }  
 }  
 return result;  
 }  
  
 public void setImagesList(){  
 if (!window\_active) {  
 SharedPreferences settings = getActivity().getSharedPreferences("Settings", Context.*MODE\_PRIVATE*);  
 workImagesList = getImagesListsFromString(settings.getString(elemsSettingsName, ""));  
 if (workImagesList != null & workImagesList.size() > 0) {  
 if (workImagesList.get(0).size() > 0) {  
 adapter = new ImagesListAdapter(getActivity(), R.layout.*images\_list*, workImagesList, getActivity());  
 listView.setAdapter(adapter);  
 } else  
 workImagesList = new ArrayList<>();  
 } else  
 workImagesList = new ArrayList<>();  
 window\_active = true;  
 }  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 window\_active =false;  
 }  
  
 @Override  
 public void onPause() {  
 super.onPause();  
 SharedPreferences settings = getActivity().getSharedPreferences("Settings", Context.*MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = settings.edit();  
 editor.putString(elemsSettingsName, imagesListsToString(workImagesList));  
 editor.apply();  
 }  
  
 @Override  
 public void onResume() {  
 super.onResume();  
 setImagesList();  
 }  
  
 @Override  
 public void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 Display screensize = getActivity().getWindowManager().getDefaultDisplay();  
 Point size = new Point();  
 screensize.getSize(size);  
 *width* = size.x;  
 height = size.y;  
 }  
  
 public View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
 root = inflater.inflate(R.layout.*fragment\_four\_tab\_images*, container, false);  
 FloatingActionButton addImageButton = root.findViewById(R.id.*imageAddBtn*);  
 listView = root.findViewById(R.id.*imagesList*);  
  
 addImageButton.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View view) {  
 Intent intent = new Intent(Intent.*ACTION\_PICK*,  
 android.provider.MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 startActivityForResult(intent, success);  
 }  
 });  
  
 return root;  
 }  
  
  
 @Override  
 public void onActivityResult(int requestCode, int resultCode, Intent imageReturnedIntent) {  
 super.onActivityResult(requestCode, resultCode, imageReturnedIntent);  
  
 if (!window\_active){  
 window\_active = true;  
 }  
  
 if (requestCode == success & imageReturnedIntent!=null) {  
 if (resultCode == *RESULT\_OK*) {  
 try {  
 final Uri imageUri = imageReturnedIntent.getData();  
 final InputStream imageStream = getContext().getContentResolver().openInputStream(imageUri);  
 Bitmap uploadImage = BitmapFactory.*decodeStream*(imageStream);  
  
 String newImageName = "image\_"+imageUri.hashCode()+".jpeg";  
  
 if (workImagesList != null){  
 if (workImagesList.size()==0){  
 List<String> tempImageList = new ArrayList<>();  
 workImagesList.add(tempImageList);  
 }  
 if (workImagesList.get(workImagesList.size()-1).size()>=8){  
 List<String> tempImageList = new ArrayList<>();  
 tempImageList.add(newImageName);  
 workImagesList.add(tempImageList);  
 }  
 else {  
 workImagesList.get(workImagesList.size()-1).add(newImageName);  
 }  
 }  
  
 ByteArrayOutputStream bos2 = new ByteArrayOutputStream();  
 if (uploadImage.getWidth() < 300 | uploadImage.getHeight() < 300)  
 uploadImage.compress(Bitmap.CompressFormat.*JPEG*, 70, bos2);  
 else {  
 float ratio = (float)uploadImage.getWidth()/uploadImage.getHeight();  
 uploadImage = Bitmap.*createScaledBitmap*(uploadImage, (int)(300\*ratio), 300, false);  
 uploadImage.compress(Bitmap.CompressFormat.*JPEG*, 70, bos2);  
 }  
  
 byte[] bitmapdata = bos2.toByteArray();  
 File imageFile = new File(getContext().getFilesDir(), newImageName);  
  
 try {  
 FileOutputStream fos = new FileOutputStream(imageFile);  
 fos.write(bitmapdata);  
 fos.flush();  
 fos.close();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 if(workImagesList != null & workImagesList.get(0).size()==1){  
 adapter = new ImagesListAdapter(getActivity(), R.layout.*images\_list*, workImagesList, getActivity());  
 listView.setAdapter(adapter);  
 }  
 else if (workImagesList.size()>0){  
 adapter.notifyDataSetChanged();  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
  
 static class ImagesListAdapter extends ArrayAdapter<List<String>> {  
 private final List<List<String>> taskImg;  
 Activity generalAct;  
  
 ImagesListAdapter(Context context, int textViewResourceId, List<List<String>> objects, Activity generalAct) {  
 super(context, textViewResourceId, objects);  
 this.taskImg = objects;  
 this.generalAct = generalAct;  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater inflater = (LayoutInflater) getContext().getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 @SuppressLint("ViewHolder") View row = inflater.inflate(R.layout.*images\_list*, parent, false);  
  
 *layout* = row.findViewById(R.id.*imageSet*);  
 ViewGroup.LayoutParams params = *layout*.getLayoutParams();  
 params.height = *width*;  
 params.width = *width*;  
 *layout*.setLayoutParams(params);  
  
 List<ImageView> imagesListToShow = new ArrayList<>();  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img1*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img2*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img3*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img4*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img5*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img6*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img7*));  
 imagesListToShow.add(row.findViewById(R.id.*gal\_img8*));  
  
 List<ProgressBar> loadingStatusList = new ArrayList<>();  
 loadingStatusList.add(row.findViewById(R.id.*load1*));  
 loadingStatusList.add(row.findViewById(R.id.*load2*));  
 loadingStatusList.add(row.findViewById(R.id.*load3*));  
 loadingStatusList.add(row.findViewById(R.id.*load4*));  
 loadingStatusList.add(row.findViewById(R.id.*load5*));  
 loadingStatusList.add(row.findViewById(R.id.*load6*));  
 loadingStatusList.add(row.findViewById(R.id.*load7*));  
 loadingStatusList.add(row.findViewById(R.id.*load8*));  
  
 int imgNumber = taskImg.get(position).size();  
  
 for (int i=0; i<8; i++){  
 try {  
 if (i<imgNumber){  
 LoadImage handler = new LoadImage(imagesListToShow.get(i), generalAct, position, getContext(), taskImg.get(position).get(i));  
 Thread th = new Thread(handler);  
 th.start();  
 }  
 else loadingStatusList.get(i).setVisibility(View.*INVISIBLE*);  
 } catch (Exception ignored){}  
 }  
  
 return row;  
 }  
  
 public class LoadImage implements Runnable {  
 protected ImageView imageView;  
 protected Activity uiActivity;  
 protected Context context;  
 protected int position;  
 protected String fileName;  
  
 public LoadImage(ImageView imageView, Activity uiActivity, int position, Context context, String fileName) {  
 this.imageView = imageView;  
 this.uiActivity = uiActivity;  
 this.context = context;  
 this.fileName = fileName;  
 this.position = position;  
 }  
  
 public void run() {  
 try {  
 File imageFile = new File(context.getFilesDir() + "/" + fileName);  
 InputStream is = new FileInputStream(imageFile);  
  
 Bitmap userImage = BitmapFactory.*decodeStream*(is);  
  
 uiActivity.runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 imageView.setImageBitmap(userImage);  
 }  
 });  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
}

**Висновок**

Виконано п’яту лабораторну роботу. Створено четверту вкладку, на якій відображаються зображення, які користувач завантажує в додаток.