LAPORAN TUGAS BESAR KELOMPOK APLIKASI KOMPUTASI BERGERAK

IF-7

Dosen : Sopian Alviana, S.Kom., M.Kom

Disusun Oleh :

Ahmad Shofi (10118273)

Mirza My Humayung (10118277)

Ahmad Umar Faruq (10118310)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA

2021

Pendahuluan

Pada zaman sekarang, hampir semua orang sudah menggunakan *smartphone*. *smartphone* sudah menjadi item wajib yang harus dimiliki karena dapat membantu kehidupan sehari-hari seperti komukasi, pekerjaan, pendidikan, hiburan dan masih banyak lagi. Maka tak heran mengapa orang-orang begitu menginginkan *smartphone* yang desain yang bagus, preferensi yang luas dan memiliki performa yang tinggi.

Berbicara mengenai preferensi, bagi sebagian besar masyarakat, hal-hal yang menjadi faktor utama salah satunya adalah *wallpaper*. *Wallpaper* merupakan gambar yang terpasang menjadi latar belakang halaman depan sebuah *smartphone*. Pemilihan Wallpaper yang bagus menjadi kunci penting yang akan menambah kenyamanan saat menggunakan *smartphone*.

Latar Belakang

Dengan melihat pentingnya sebuah *wallpaper* pada sebuah *smartphone*, maka pemilihannya juga perlu diperhatikan. Apakah *wallpaper* yang dipilih cocok dengan selera? Apakah *wallpaper* tersebut menambah nilai keindahan smartphone? Namun pemilihan wallpaper yang cocok, tidaklah semudah yang dibayangkan. Karena kita harus mencari gambar yang cocok dari berbagai sumber seperti, **Pinterest**, **Instagram**, **Google**, **PixaBay**, **Unsplash** dan lain sebagainya. Maka dari itu, kami berusaha menemukan solusi yang tepat untuk menangani masalah ini.

Jika kita membahas mengenai penyedia gambar yang berpotensi untuk dijadikan wallpaper, ada banyak sumber-sumber bagus seperti yang telah disebutkan di paragraf sebelumnya. Kita ambil satu contoh, yaitu Pixabay.

Dikutip dari **Pixabay.com**:

“Pixabay.com is a website for sharing photos, illustrations, vector graphics, film footage and music, exclusively under the custom Pixabay license, which generally allows the free use of the material with some restrictions.”[1][2][3]

Terjemahan:

“Pixabay.com adalah situs web untuk berbagi foto, ilustrasi, grafik vektor, cuplikan film, dan musik, secara eksklusif di bawah lisensi kustom Pixabay, yang umumnya mengizinkan penggunaan materi secara gratis dengan beberapa batasan.”[1][2][3]

Bisa disimpulkan dari kutipan diatas bahwa **pixabay** berpotensi untuk dijadikan sumber **wallpaper** dari banyak orang didunia secara gratis. Namun **pixabay.com** adalah sebuah *website*. Walaupun ada untuk versi mobilenya, namun akan lebih bagus jika ada sebuah klien yang akan mengambil gambar langsung dari pixabay.com dengan sajian yang berbeda, melalui API (Application Program Interface) yang secara resmi disediakan oleh pixabay.com sendiri secara gratis.

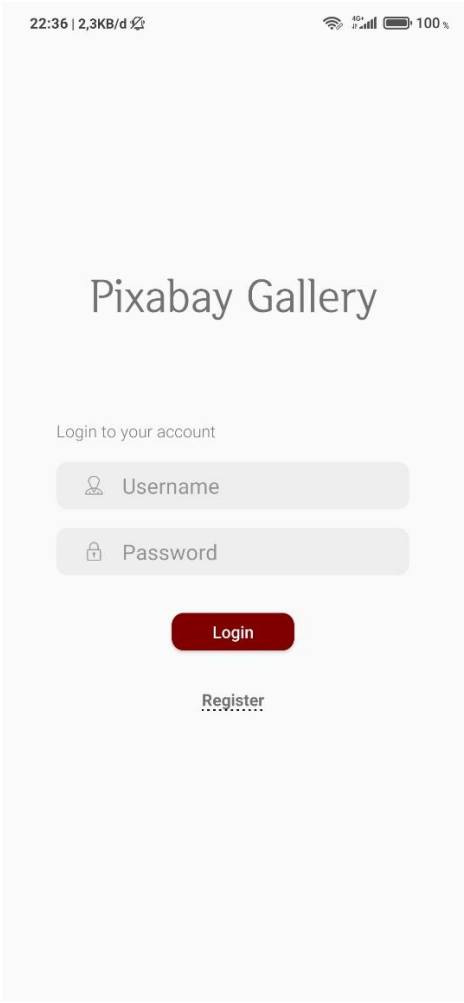
Penjelasan Aplikasi

Aplikasi klient mobile yang akan kami buat bernama “Pixabay Gallery”. Aplikasi ini berguna untuk mengambil data gambar dari pixabay.com melalui API, lalu menampilkannya ke antarmuka dengan tambahan beberapa fitur. Denga aplikasi ini, pengguna dapat dengan mudah mencari gambar yang diinginkan karena terdapat fitu pencarian. Aplikasi ini akan menampilkan gambar berdasarkan kata kunci yang diberikan. Jika pengguna memberikan kata kunci “pantai”, maka hasil yang akan ditampilkan adalah gambar-gambar yang berhubungan denga pantai. Jika pengguna memberikan kata kunci “langit, maka aplikasi akan menampilkan gambar-gambar yang berhubungan dengan langit.

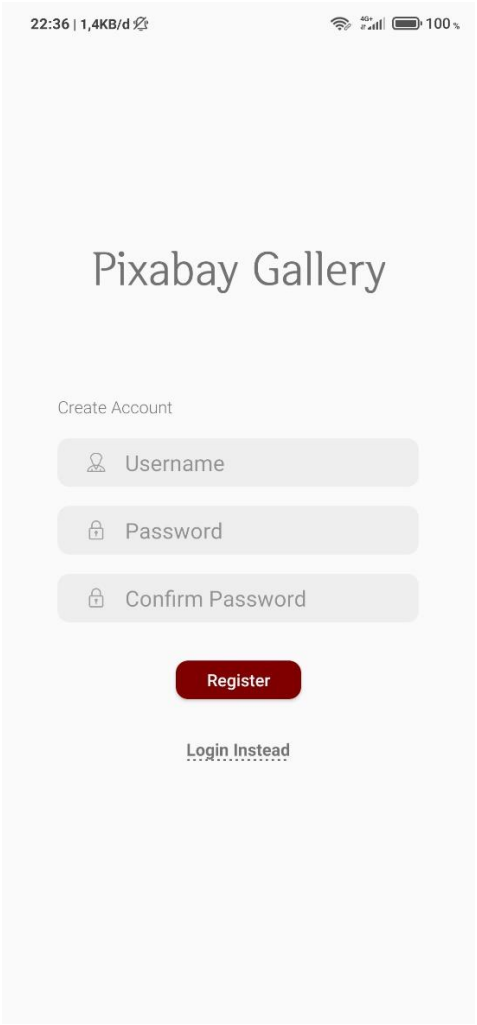
Aplikasi ini juga dibekali fitur “Set As”, dimana jika pengguna mengklik tombol tersebut, maka secara instan pengguna bisa langsung menjadikan gambar tersebut sebagai wallpaper. Fitur “Add To Favorite” untuk meyimpan gambar favorite untuk dilihat nantinya.

Lampiran Aplikasi

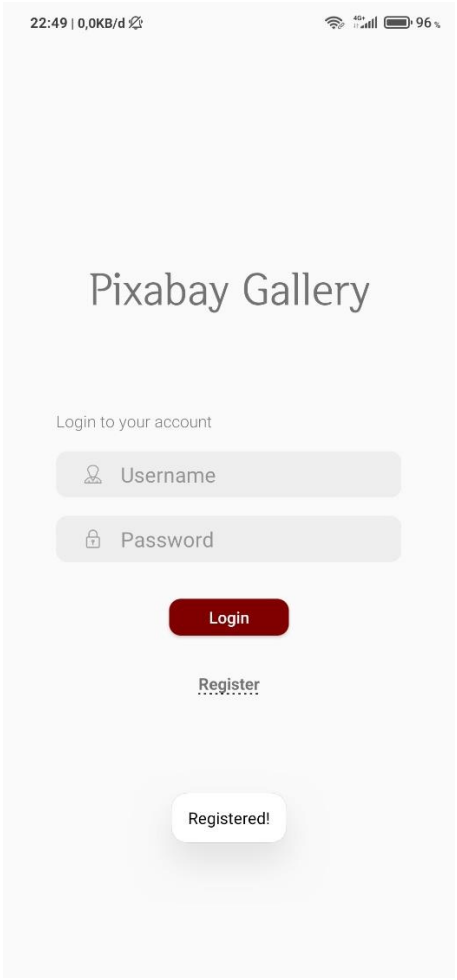
1. Pada saat pengguna membuka aplikasi, maka langsung ke menu halaman login. Pengguna juga bisa langsung mengisikan username dan password jika sudah terdaftar. Tapi jika belum, pengguna bisa mendaftar dengan cara mengklik opsi “Register” yang terdapat dibawah tombol login.



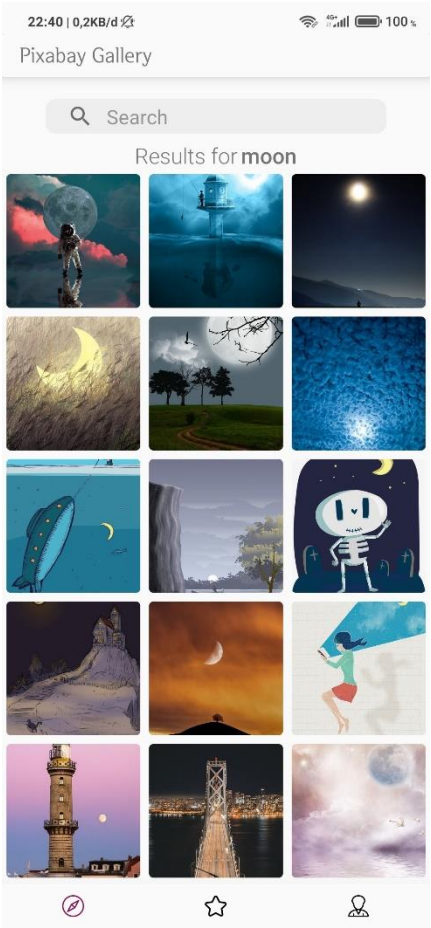
2. Setelah berada di halaman Register, pengguna bisa memasukkan username dan password untuk mendaftar. Password diisikan dua kali agar pengguna tidak salah dalam pengetikan password. Lalu klik tombol “Register” untuk mendaftar



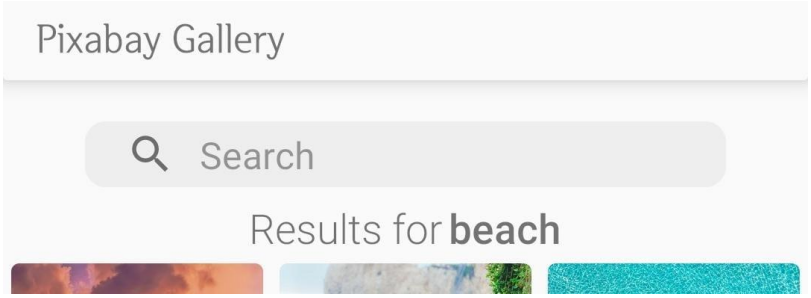
3. Setelah mengklik tombol “Register” pengguna dialihkan ke halaman login dan menerima pesan pop-up berisikan “Registered!”. Lalu isikan username dan password yang sudah didaftarkan tadi untuk login.



4. Setelah login, pengguna langsung dialihkan ke halaman explore. Pada menu explore, terdapat banyak gambar yang ditampilkan. Pada bagian bawah terdapat 3 opsi untuk menuju ke halaman explore (kiri), halaman favorite (tengah), dan halaman akun (kanan).



5. Menu explore terdapat searchbox untuk mencari kriteria gambar sesuai dengan pengguna inginkan.



6. Saat pengguna mengklik salah satu gambar, gambar akan secara otomatis terbuka,

Dan terdapat 3 opsi, yaitu :

1. Favourite : berfungsi untuk menyimpan gambar ke halaman favourite, supaya pegguna bisa lebih gambar menemukan gambar yang dia suka.
2. Download : Berfungsi untuk mendownload/meyimpan gambar ke perangkat.
3. Set As : Pada opsi ini berfungsi untuk meng-set gambar yang pengguna lihat tadi, bisa digunakan untuk foto profile whatsapp dan kontak telepon. Juga bisa diset sebagai wallpaper smartphone pengguna.

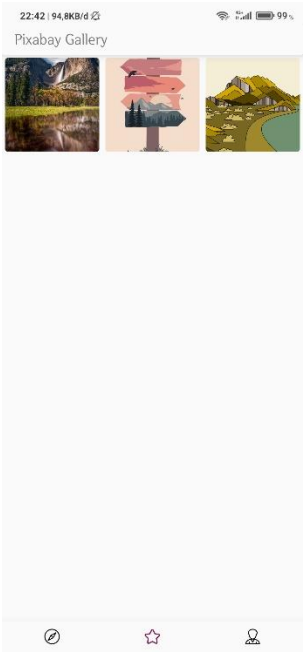
Selain 3 opsi tersebut, juga terdapat info mengenai gambar, seperti likes, tags, resolution, size, views, downloads, dan comments.

7. Saat pengguna meng set as gambar sebagai wallpaper smartphone, maka tampilannya akan seperti ini.

Setelah memilih Set As ke “Walpaper”, maka akan tampil seperti ini. Sesuaikan potongan gambar sesuai dengan view dari foto.

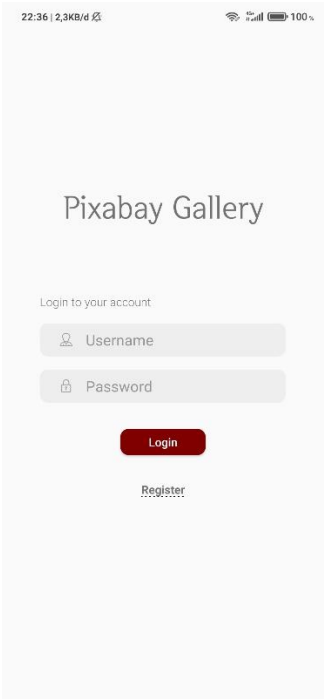
Setelah mengklik “Selesai”, Kembali ke halaman awal smartphone, dengan hasil seperti dibawah:

8. Pada menu favourite, terdapat gambar yang pengguna save sebagai favourite tadi.



9. Terakhir, pada halaman akun, dimana halaman akun ini terdapat foto profile pengguna dan nama pengguna. Serta ada opsi “Logout” untuk keluar dari akun.

10. Setelah mengklik tombol “Logout” maka akan kembali ke halaman login.



Penjelasan Fungsionalitas

Penjelasan fungsionalitas Aplikasi

Ada beberapa fungsionalitas pada aplikasi ini:
1. bisa mendowload file foto yang ada pada aplikasi ini
2. bisa melakukan setAs foto di aplikasi ini
3. memfavoritkan gambar

Menu

Menu pada aplikasi pixabay ini ada 3 diantara nya:

1. Menu explorer : menu ini berfungsi untuk menampilkan gambar default pada aplikasi ini, di menu explorer kita juga bisa mencari gambar yang kita ngingikan dengan memasukkan kata kunci pada kolom search

2. Menu favorite : menu favourite ini berfungsi ketika pada halaman home kita mengklik sebuah gambar maka akan beralir ke halaman detail, nah ketika kita mengklik icon favourite maka ketika kita beralih ke halaman favourite aka nada gambar yang sudah kita favouritekan.
3. Menu Account : pada menu account ini user bisa melakukan login dan registrasi akun pada aplikasi pixabay ini

Pembuatan (Code)Aplikasi

ExploreAdapter.kt

```
package com.example.pixabayimages.adapter
```

```kotlin
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.RecyclerView
import androidx.swiperefreshlayout.widget.CircularProgressDrawable
import com.bumptech.glide.Glide
import com.example.pixabayimages.R
import com.example.pixabayimages.TAG
import com.example.pixabayimages.model.PixabayResponse
import kotlinx.android.synthetic.main.item_explore.view.*
import org.koin.core.KoinComponent


class ExploreAdapter(
    private var data: ArrayList<PixabayResponse.Photo>,
    var context: AppCompatActivity
) :
        RecyclerView.Adapter<ExploreAdapter.Holder>(), KoinComponent {
    private val maxPlaceHolder = 8
    var onItemClick: (data: PixabayResponse.Photo, position: Int) -> Unit = { _, _
-> }
    var onLongClick: (data: PixabayResponse.Photo, position: Int) -> Unit = { _, _
-> }
    var lastPosition: Int = -1
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Holder {
        return Holder(
            LayoutInflater
                .from(context)
                .inflate(R.layout.item_explore, parent, false)
        )
    }

    override fun getItemCount(): Int {
        return data.size
    }

    fun getData(): ArrayList<PixabayResponse.Photo> {
        return data
    }

    fun replace(list: ArrayList<PixabayResponse.Photo>) {
        this.data = list
        notifyDataSetChanged()
    }

    fun clear() {
        this.data.clear()
        notifyDataSetChanged()
    }

    fun setLoading() {
        (1..maxPlaceHolder).forEach { _ ->
            this.data.add(PixabayResponse.Photo())
        }
        notifyDataSetChanged()
```

```
    }

    fun removeLoading() {
        val loadingCount = this.data.count { it.webformatURL == "" }
        (1..loadingCount).forEach { _ ->
            val lastIndex = this.data.size - 1
            if (this.data[lastIndex].webformatURL == "") {
                this.data.removeAt(lastIndex)
            }
        }
        notifyDataSetChanged()
    }

    fun addAll(list: ArrayList<PixabayResponse.Photo>) {
        this.data.addAll(list)
        notifyDataSetChanged()
    }

    override fun onViewDetachedFromWindow(holder: Holder) {
        super.onViewDetachedFromWindow(holder)
        holder.itemView.clearAnimation()
    }

    override fun onBindViewHolder(holder: Holder, position: Int) {
        if(position == 0){
            Log.d(TAG, "Hi im first item from the explore list! ")
            Log.d(TAG, "This is my imageURL: ${data[position].webformatURL}")
        }
        val drw = CircularProgressDrawable(context)
        drw.strokeWidth = 5f
        drw.centerRadius = 30f
        drw.start()
        holder.itemView.image.setImageDrawable(null)
        Glide.with(context)
            .load(
                data[position]
                    .webformatURL
            )
            .placeholder(drw)
            .centerCrop()
            .into(holder.itemView.image)

        holder.itemView.setOnClickListener {
            onItemClick(data[position], position)
        }

        holder.itemView.setOnLongClickListener {
            onLongClick(data[position], position)
            true
        }

        val animation: Animation = AnimationUtils.loadAnimation(context, if
(position > lastPosition) R.anim.up_from_bottom else R.anim.down_from_top)
        holder.itemView.startAnimation(animation)
        lastPosition = position

    }
```

```
    inner class Holder(itemView: View) : RecyclerView.ViewHolder(itemView){

    }
}
```

FavouriteAdapter.kt

```
package com.example.pixabayimages.adapter

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.RecyclerView
import androidx.swiperefreshlayout.widget.CircularProgressDrawable
import com.bumptech.glide.Glide
import com.example.pixabayimages.R
import com.example.pixabayimages.model.PixabayResponse
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.item_explore.view.*
import org.koin.core.KoinComponent
import org.koin.core.inject

class FavouriteAdapter(private var data: ArrayList<PixabayResponse.Photo?>, var
context:
AppCompatActivity):RecyclerView.Adapter<FavouriteAdapter.Holder>(),KoinComponent {
    private val maxPlaceHolder = 8
    var onItemClick: (data: PixabayResponse.Photo, position: Int) -> Unit = { _, _
-> }
    var onLongClick: (data: PixabayResponse.Photo, position: Int) -> Unit = { _, _
-> }
    var lastPosition: Int = -1
    private val preferences: Preferences by inject()




    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
FavouriteAdapter.Holder {
        return Holder(
            LayoutInflater
                .from(context)
                .inflate(R.layout.item_explore, parent, false)
        )
    }

    override fun getItemCount(): Int {
        return data.size
    }

    fun getData(): ArrayList<PixabayResponse.Photo?> {
        return data
    }

    fun replace(list: ArrayList<PixabayResponse.Photo?>) {
```

```
        this.data = list
        notifyDataSetChanged()
    }

    fun clear() {
        this.data.clear()
        notifyDataSetChanged()
    }

//    fun setLoading() {
//        (1..maxPlaceHolder).forEach { _ ->
//            this.data.add(PixabayResponse.Photo())
//        }
//        notifyDataSetChanged()
//    }
//
//    fun removeLoading() {
//        val loadingCount = this.data.count { it.webformatURL == "" }
//        (1..loadingCount).forEach { _ ->
//            val lastIndex = this.data.size - 1
//            if (this.data[lastIndex].webformatURL == "") {
//                this.data.removeAt(lastIndex)
//            }
//        }
//        notifyDataSetChanged()
//    }

//    fun addAll(list: ArrayList<PixabayResponse.Photo>) {
//        this.data.addAll(list)
//        notifyDataSetChanged()
//    }

    override fun onViewDetachedFromWindow(holder: FavouriteAdapter.Holder) {
        super.onViewDetachedFromWindow(holder)
        holder.itemView.clearAnimation()
    }

    override fun onBindViewHolder(holder: Holder, position: Int) {
//        if(position == 0){
//            Log.d(TAG, "Hi im first item from the explore list! ")
//            Log.d(TAG, "This is my imageURL: ${data[position].webformatURL}")
//        }


        val drw = CircularProgressDrawable(context)
        drw.strokeWidth = 5f
        drw.centerRadius = 30f
        drw.start()
        holder.itemView.image.setImageDrawable(null)
        Glide.with(context)
            .load(
                data[position]
                    ?.webformatURL

            )
            .placeholder(drw)
            .centerCrop()
            .into(holder.itemView.image)
```

```
        holder.itemView.setOnClickListener {
            onItemClick(data[position]!!, position)
        }

        holder.itemView.setOnLongClickListener {
            onLongClick(data[position]!!, position)
            true
        }

        val animation: Animation = AnimationUtils.loadAnimation(context, if
(position > lastPosition) R.anim.up_from_bottom else R.anim.down_from_top)
        holder.itemView.startAnimation(animation)
        lastPosition = position

    }
    fun setData(data: ArrayList<PixabayResponse.Photo?>){
        this.data = data
    }



    inner class Holder(itemView: View) : RecyclerView.ViewHolder(itemView){

    }
}
```

ViewPagerAdapter.kt

```
package com.example.pixabayimages.adapter

import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentActivity
import androidx.viewpager2.adapter.FragmentStateAdapter

class ViewPagerAdapter(fa: FragmentActivity) : FragmentStateAdapter(fa) {
    private val mFragmentList: MutableList<Fragment> = ArrayList()
    fun clear() {
        mFragmentList.clear()
        notifyDataSetChanged()
    }

    fun add(fragment: Fragment, title: String) {
        mFragmentList.add(fragment)
        notifyDataSetChanged()
    }

    fun replaceAll(fragments: ArrayList<Fragment>) {
        mFragmentList.clear()
        mFragmentList.addAll(fragments)
        notifyDataSetChanged()
    }

    override fun getItemCount(): Int {
        return mFragmentList.size
    }

    override fun createFragment(position: Int): Fragment {
        return mFragmentList[position]
```

```
    }

    fun toggle(order: Int): Boolean {
        return true
    }
}
```

ConstrainImageView.kt

```
package com.example.pixabayimages.customView

import android.content.Context
import android.util.AttributeSet

class ConstraintImageView : androidx.appcompat.widget.AppCompatImageView {
    constructor(context: Context?) : super(context!!) {}
    constructor(context: Context?, attrs: AttributeSet?) : super(context!!, attrs)
{}
    constructor(context: Context?, attrs: AttributeSet?, defStyle: Int) :
super(context!!, attrs, defStyle) {}

    override fun onMeasure(widthMeasureSpec: Int, heightMeasureSpec: Int) {

        super.onMeasure(widthMeasureSpec, heightMeasureSpec) // This is the key
that will make the height equivalent to its width
    }
}
```

SquareImageView.kt

```
package com.example.pixabayimages.customView

import android.content.Context
import android.util.AttributeSet
import android.widget.ImageView

class SquareImageView : androidx.appcompat.widget.AppCompatImageView {
    constructor(context: Context?) : super(context!!) {}
    constructor(context: Context?, attrs: AttributeSet?) : super(context!!, attrs)
{}
    constructor(context: Context?, attrs: AttributeSet?, defStyle: Int) :
super(context!!, attrs, defStyle) {}

    override fun onMeasure(widthMeasureSpec: Int, heightMeasureSpec: Int) {
        super.onMeasure(widthMeasureSpec, widthMeasureSpec) // This is the key
that will make the height equivalent to its width
    }
}
```

FragmentAccount.kt

```
package com.example.pixabayimages.fragment

import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
```

```kotlin
import android.view.ViewGroup
import androidx.core.content.res.ResourcesCompat
import androidx.fragment.app.Fragment
import com.example.pixabayimages.MemoryDb
import com.example.pixabayimages.R
import com.example.pixabayimages.TAG
import com.example.pixabayimages.model.UserData
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.fragment_account.*
import kotlinx.android.synthetic.main.fragment_details.*
import kotlinx.android.synthetic.main.fragment_details.userName
import org.koin.android.ext.android.inject


// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [FragmentAccount.newInstance] factory method to
 * create an instance of this fragment.
 */
class FragmentAccount : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null
    private val preferences: Preferences by inject()
    private val memoryDb: MemoryDb by inject()



    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_account, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        btn_logout.setOnClickListener {
            memoryDb.currentUser.value = UserData()
        }
        Log.d(TAG, "onViewCreated: ${preferences.getCurrentUser()}")
        val currentUser = preferences.getCurrentUser()
        userName.text = currentUser?.username

        super.onViewCreated(view, savedInstanceState)
    }
```

```
    public fun setUsername(uname: String){
        userName.text = uname
    }

    companion object {
        /**
         * Use this factory method to create a new instance of
         * this fragment using the provided parameters.
         *
         * @param param1 Parameter 1.
         * @param param2 Parameter 2.
         * @return A new instance of fragment AccountFragment.
         */
        // TODO: Rename and change types and number of parameters
        @JvmStatic
        fun newInstance(param1: String, param2: String) =
            FragmentAccount().apply {
                arguments = Bundle().apply {
                    putString(ARG_PARAM1, param1)
                    putString(ARG_PARAM2, param2)
                }
            }
    }
}
```

FragmentDetails.kt

```
package com.example.pixabayimages.fragment

import android.annotation.SuppressLint
import android.app.Dialog
import android.app.DownloadManager
import android.content.Context.DOWNLOAD_SERVICE
import android.content.Intent
import android.database.Cursor
import android.graphics.drawable.Drawable
import android.net.Uri
import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.core.content.FileProvider
import com.bumptech.glide.Glide
import com.example.pixabayimages.MemoryDb
import com.example.pixabayimages.R
import com.example.pixabayimages.TAG
import com.example.pixabayimages.model.PixabayResponse
import com.example.pixabayimages.model.UserData
import com.example.pixabayimages.persistence.Preferences
import com.example.pixabayimages.toMb
import com.google.android.material.bottomsheet.BottomSheetDialogFragment
import kotlinx.android.synthetic.main.fragment_details.*
import kotlinx.coroutines.*
import org.koin.android.ext.android.inject
import java.io.File
```

```
class FragmentDetails : BottomSheetDialogFragment() {
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
    var a : Int = 0
    var onDismissed: (data: String) -> Unit = { _ -> }
    var photo : PixabayResponse.Photo? = null
    var preview : Drawable? = null

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {

        return inflater.inflate(R.layout.fragment_details, container, false)
    }

    fun setData(photo : PixabayResponse.Photo, preview : Drawable){
        this.photo = photo
        this.preview = preview
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        setDetails()
        setAs.setOnClickListener {
            setAs()
        }
        favorite.setOnClickListener {
            setLikes()
        }
        super.onViewCreated(view, savedInstanceState)
    }

    private fun setDetails(){
        photo?.let {
            val res = getString(R.string.resolution_val, it.imageWidth,
it.imageHeight)
            val fileSize = toMb(it.imageSize.toInt())
            val formattedFileSize = getString(R.string.size_val, fileSize)

            userName.text = it.user
            likes.text = it.likes
            tags.text = it.tags
            size.text = formattedFileSize
            resolution.text = res
            views.text = it.views
            downloads.text = it.downloads
            comments.text = it.comments

            Glide.with(requireContext())
                .load(it.userImageURL)
                .placeholder(R.drawable.ic_person_foreground)
                .into(pfp)
            Glide.with(requireContext())
                .load(it.largeImageURL)
                .placeholder(preview)
```

```
                .into(largeImage)
        }
    }


    private fun setLikes(){
        val currentUser = preferences.getCurrentUser()
        val favoriteList = currentUser?.favoriteList
        favoriteList?.add(photo!!)
        preferences.setCurrentUser(currentUser!!)
        Log.d(TAG, "setLikes: ${currentUser.favoriteList.size}")
        updateUserData(currentUser)
    }

    private fun updateUserData(userData : UserData) : Boolean {
        val userList = preferences.getUserList()
        val length = userList.size - 1
        (0..length).forEach {
            val data = userList[it]
            if(data?.username == userData.username){
                userList[it] = userData
                preferences.setUserList(userList)
                return true
            }
        }

        return false
    }

    private fun setAs(){
        val url = photo!!.largeImageURL
        val context = requireContext()
        val rm = Glide.with(this).downloadOnly().load(url).submit()
        GlobalScope.launch(Dispatchers.Main, CoroutineStart.DEFAULT) {
            val defFile: Deferred<File> = async(Dispatchers.IO) { rm.get() }
            val file = defFile.await()
            val intent = Intent(Intent.ACTION_ATTACH_DATA)
            val uri = FileProvider.getUriForFile(
                context,
                context.packageName + ".provider",
                file
            )
            intent.flags = Intent.FLAG_GRANT_READ_URI_PERMISSION
            intent.addCategory(Intent.CATEGORY_DEFAULT)
            intent.setDataAndType(uri, "image/jpeg")
            intent.putExtra("mimeType", "image/jpeg")
            startActivity(Intent.createChooser(intent, "Set as:"))

        }
    }


    // TODO: fix this
    private fun beginDownload() {
        val url = "http://speedtest.ftp.otenet.gr/files/test10Mb.db"
        var fileName = url.substring(url.lastIndexOf('/') + 1)
        fileName = fileName.substring(0, 1).toUpperCase() + fileName.substring(1)
//        val file: File = Util.DocumentFile(fileName, context)
        val request = DownloadManager.Request(Uri.parse(url))
```

```
            .setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE)
// Visibility of the download Notification
//            .setDestinationUri(Uri.fromFile(file)) // Uri of the destination
file
            .setTitle(fileName) // Title of the Download Notification
            .setDescription("Downloading") // Description of the Download
Notification
            .setRequiresCharging(false) // Set if charging is required to begin
the download
            .setAllowedOverMetered(true) // Set if download is allowed on Mobile
network
            .setAllowedOverRoaming(true) // Set if download is allowed on roaming
network
        val downloadManager = activity?. getSystemService(DOWNLOAD_SERVICE) as
DownloadManager?
        val downloadID =
            downloadManager!!.enqueue(request) // enqueue puts the download
request in the queue.

        // using query method
        var finishDownload = false
        var progress: Int
        while (!finishDownload) {
            val cursor: Cursor =

downloadManager.query(DownloadManager.Query().setFilterById(downloadID))
            if (cursor.moveToFirst()) {
                val status: Int =

cursor.getInt(cursor.getColumnIndex(DownloadManager.COLUMN_STATUS))
                when (status) {
                    DownloadManager.STATUS_FAILED -> {
                        finishDownload = true
                    }
                    DownloadManager.STATUS_PAUSED -> {
                    }
                    DownloadManager.STATUS_PENDING -> {
                    }
                    DownloadManager.STATUS_RUNNING -> {
                        val total: Long =

cursor.getLong(cursor.getColumnIndex(DownloadManager.COLUMN_TOTAL_SIZE_BYTES))
                        if (total >= 0) {
                            val downloaded: Long =

cursor.getLong(cursor.getColumnIndex(DownloadManager.COLUMN_BYTES_DOWNLOADED_SO_FA
R))
                            progress = (downloaded * 100L / total).toInt()
                            // if you use downloadmanger in async task, here you
can use like this to display progress.
                            // Don't forget to do the division in long to get more
digits rather than double.
                            //  publishProgress((int) ((downloaded * 100L) /
total));
                        }
                    }
                    DownloadManager.STATUS_SUCCESSFUL -> {
                        progress = 100
                        // if you use aysnc task
```

```
                        // publishProgress(100);
                        finishDownload = true
                        Toast.makeText(activity, "Download Completed",
Toast.LENGTH_SHORT)
                            .show()
                    }
                }
            }
        }
    }


    @SuppressLint("RestrictedApi")
    override fun setupDialog(dialog: Dialog, style: Int) {
        super.setupDialog(dialog, style)
    }


}
```

FragmentExplorer.kt

```
package com.example.pixabayimages.fragment

import android.annotation.SuppressLint
import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.pixabayimages.MemoryDb
import com.example.pixabayimages.R
import com.example.pixabayimages.adapter.ExploreAdapter
import com.example.pixabayimages.networking.ApiReq
import com.example.pixabayimages.networking.Status
import kotlinx.android.synthetic.main.fragment_explore.*
import org.koin.android.ext.android.inject
import androidx.appcompat.widget.SearchView.OnQueryTextListener
import androidx.core.view.get
import com.example.pixabayimages.PER_PAGE
import com.example.pixabayimages.TAG
import com.example.pixabayimages.model.ImageDetails
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.item_explore.view.*


class FragmentExplore : Fragment() {
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
    private val apiReq: ApiReq by inject()
    private var page: Int = 1
    private var loading = false
    private var isTheEnd = false
    private val fragmentDetails: FragmentDetails by inject()
```

```
    private lateinit var exploreAdapter: ExploreAdapter

    @SuppressLint("NewApi")
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        exploreAdapter = ExploreAdapter(arrayListOf(), requireContext() as
AppCompatActivity)
        exploreAdapter.onItemClick = fun(model, position) {
            val previewImage =
list.findViewHolderForAdapterPosition(position)?.itemView?.image
            fragmentDetails.setData(model, previewImage!!.drawable)
            fragmentDetails.show(childFragmentManager, "Details")

        }
        list.adapter = exploreAdapter
        resetList()
        list.addOnScrollListener(object : RecyclerView.OnScrollListener() {
            override fun onScrollStateChanged(recyclerView: RecyclerView,
newState: Int) {
                super.onScrollStateChanged(recyclerView, newState)
                val linearLayoutManager = recyclerView.layoutManager as
LinearLayoutManager?
                if (!loading && !isTheEnd) {
                    if (exploreAdapter.itemCount - 1 ==
linearLayoutManager?.findLastCompletelyVisibleItemPosition()) {
                        page++
                        loadData()
                    }
                }
            }
        })
        searchView.setOnQueryTextListener(object : OnQueryTextListener{
            override fun onQueryTextSubmit(query: String?): Boolean {
                memoryDb.query.value = query
                return query.isNullOrEmpty()
            }
            override fun onQueryTextChange(newText: String?): Boolean{
                return newText.isNullOrEmpty()
            }
        })
        refresh.setOnRefreshListener {
            resetList()
            refresh.isRefreshing = false
        }



        exploreAdapter.registerAdapterDataObserver(object :
RecyclerView.AdapterDataObserver(){
            override fun onChanged() {
                if(exploreAdapter.itemCount == 0){
                    noResultsFor.visibility = View.VISIBLE
                    resultsFor.visibility = View.GONE
                    list.visibility = View.GONE
                }else{
                    noResultsFor.visibility = View.GONE
                    resultsFor.visibility = View.VISIBLE
                    list.visibility = View.VISIBLE
                }
```

```
                query.text = preferences.getQuery()
                super.onChanged()
            }
        })
        memoryDb.query.observe(viewLifecycleOwner, {
            preferences.setQuery(searchView.query.toString())
            resetList()
            loadData()
        })


    }

    private fun resetList() {
        page = 1
        exploreAdapter.clear()
        loadData()

    }

    fun loadData() {
        loading = true
        exploreAdapter.setLoading()
        val searchQuery = preferences.getQuery()
        apiReq.getImages(page, searchQuery)
            .observe(viewLifecycleOwner, {
                exploreAdapter.removeLoading()
                if (it.status == Status.SUCCESS) {
                    exploreAdapter.addAll(it.response!!.hits)
                    isTheEnd = it.response.hits.size < PER_PAGE
                }
                loading = false
            }
        )
    }


    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_explore, container, false)
    }


}
```

FragmentFavorite.kt

```
package com.example.pixabayimages.fragment

import android.os.Bundle
import android.util.Log
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
```

```
import androidx.appcompat.app.AppCompatActivity
import com.example.pixabayimages.MemoryDb
import com.example.pixabayimages.R
import com.example.pixabayimages.TAG
import com.example.pixabayimages.adapter.FavouriteAdapter
import com.example.pixabayimages.networking.ApiReq
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.fragment_explore.*
import kotlinx.android.synthetic.main.fragment_favorite.*
import kotlinx.android.synthetic.main.item_explore.view.*
import org.koin.android.ext.android.inject


class FragmentFavorite : Fragment() {
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
    private val apiReq: ApiReq by inject()
    private var page: Int = 1
    private var loading = false
    private var isTheEnd = false
    private val fragmentDetails: FragmentDetails by inject()
    private lateinit var favouriteAdapter: FavouriteAdapter


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {

        }
    }


    override fun onResume() {
        val currentUser = preferences.getCurrentUser()
        favouriteAdapter.setData(currentUser!!.favoriteList)
        favouriteAdapter.notifyDataSetChanged()
        super.onResume()
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        val currentUser = preferences.getCurrentUser()
        favouriteAdapter = FavouriteAdapter(currentUser!!.favoriteList,
requireContext() as AppCompatActivity)
        favouriteAdapter.onItemClick = fun(model, position) {
            val previewImage =
listfavourite.findViewHolderForAdapterPosition(position)?.itemView?.image
            Log.d(TAG, "im clicked: sd")
            fragmentDetails.setData(model, previewImage!!.drawable)
            fragmentDetails.show(childFragmentManager, "Details")
        }
        listfavourite.adapter = favouriteAdapter
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
```

```
        return inflater.inflate(R.layout.fragment_favorite, container, false)
    }


}
```

ImageDetails.kt

```
package com.example.pixabayimages.model

import android.graphics.drawable.Drawable

class ImageDetails(val previewImage: Drawable?, val photo : PixabayResponse.Photo)
{
}
```

PixabayResponse.kt

```
package com.example.pixabayimages.model

import java.io.Serializable

class PixabayResponse(
    var total: Int = 0,
    var totalHits: Int = 0,
    var hits: ArrayList<Photo> = arrayListOf()) : Serializable{
    class Photo {
        var id: String = ""
        var pageURL: String = ""
        var type : String = ""
        var tags : String = ""
        var previewURL = ""
        var previewWidth = ""
        var previewHeight = ""
        var webformatURL = ""
        var webformatWidth = ""
        var webformatHeight = ""
        var largeImageURL = ""
        var imageWidth = ""
        var imageHeight = ""
        var imageSize =  ""
        var views = ""
        var downloads = ""
        var collections = ""
        var likes = ""
        var comments =  ""
        var user_id = ""
        var user = ""
        var userImageURL = ""
    }
}
```

UserData.kt

```
package com.example.pixabayimages.model

class UserData (
```

```
    var username: String = "",
    var password: String = "",
    val favoriteList: ArrayList<PixabayResponse.Photo?> = arrayListOf()
)
```

Api.kt

```
package com.example.pixabayimages.networking

import android.graphics.drawable.GradientDrawable
import com.example.pixabayimages.PER_PAGE
import com.example.pixabayimages.model.PixabayResponse
import retrofit2.http.GET
import retrofit2.http.Query

interface Api {
//    @GET("api/?key=10841180-27d5e7ab760ca8594507d1a55")
//    suspend fun ping(@Query("q") random: Int): PublicImage
//
//    @GET("public/image/list")
//    suspend fun getImages(
//            @Query("page") page: Int = 1,
//            @Query("tags") tags: String = "beach"
//    ): PublicImage

    @GET("api/?key=10841180-27d5e7ab760ca8594507d1a55")
    suspend fun getImages(
            @Query("page") page: Int = 1,
            @Query("q") tags: String = "beach",
            @Query("per_page") per_page: Int = PER_PAGE,
            @Query("orientation") orientation: String = "vertical"

    ): PixabayResponse
}
```

ApiReq.kt

```
package com.example.pixabayimages.networking

import androidx.lifecycle.LiveData
import androidx.lifecycle.liveData
import com.example.pixabayimages.model.PixabayResponse
import kotlinx.coroutines.Dispatchers
import org.koin.core.KoinComponent
import org.koin.core.inject

class ApiReq : KoinComponent {
    private val repository: Repository by inject()
//    fun ping(random:Int): LiveData<Resource<PublicImage>> {
//        return liveData(Dispatchers.IO) {
//            emit(Resource.loading(null))
//            emit(repository.ping(random))
//        }
//    }

    fun getImages(page: Int, tags: String): LiveData<Resource<PixabayResponse>> {
        return liveData(Dispatchers.IO) {
```

```
            emit(Resource.loading(null))
            emit(repository.getImages(page, tags))
        }
    }
}
```

Repository.kt

```
package com.example.pixabayimages.networking

import com.example.pixabayimages.PER_PAGE
import com.example.pixabayimages.model.PixabayResponse
import org.koin.core.KoinComponent


class Repository(private val api: Api, private val responseHandler:
ResponseHandler) :
    KoinComponent {
//    suspend fun ping(random:Int): Resource<PublicImage> {
//        return try {
//            val bar = api.ping(random)
//            responseHandler.handleResponse(bar)
//        } catch (e: Exception) {
//            responseHandler.handleException(e)
//        }
//    }

    suspend fun getImages(page: Int, tags: String): Resource<PixabayResponse> {
        return try {
            val bar = api.getImages(page, tags, PER_PAGE)
            responseHandler.handleResponse(bar)
        } catch (e: Exception) {
            responseHandler.handleException(e)
        }
    }
}
```

Resource.kt

```
package com.example.pixabayimages.networking

data class Resource<out T>(val status: Status, val response: T?, val message:
String?) {
    companion object {
        fun <T> success(data: T?): Resource<T> {
            return Resource(Status.SUCCESS, data, null)
        }

        fun <T> error(msg: String?, data: T?): Resource<T> {
            return Resource(Status.ERROR, data, msg)
        }

        fun <T> loading(data: T?): Resource<T> {
            return Resource(Status.LOADING, data, null)
        }
    }
}
```

ResponHandler.kt

```kotlin
package com.example.pixabayimages.networking

import android.util.Log
import org.koin.core.KoinComponent

open class ResponseHandler : KoinComponent {

    fun <T : Any> handleResponse(data: T): Resource<T> {
        return Resource.success(data)
    }

    fun <T : Any> handleException(e: Exception): Resource<T> {
        e.printStackTrace()
        Log.d("exception", "asaa" + e.message)
        return Resource.error(e.message, null)
    }
}

class CustomException(message: String?) : java.lang.Exception(message)
```

RetrofitClient.kt

```kotlin
package com.example.pixabayimages.networking

import com.example.pixabayimages.BASE_URL
import okhttp3.OkHttpClient
import okhttp3.logging.HttpLoggingInterceptor
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

fun provideRetrofit(okHttpClient: OkHttpClient): Retrofit {
    return Retrofit.Builder().baseUrl(BASE_URL).client(okHttpClient)
        .addConverterFactory(GsonConverterFactory.create()).build()
}

fun provideOkHttpClient(loggingInterceptor: HttpLoggingInterceptor): OkHttpClient
{
    return OkHttpClient().newBuilder().addInterceptor(loggingInterceptor).build()
}

fun provideLoggingInterceptor(): HttpLoggingInterceptor {
    val logger = HttpLoggingInterceptor()
    logger.level = HttpLoggingInterceptor.Level.BODY
    return logger
}

fun provideForecastApi(retrofit: Retrofit): Api = retrofit.create(Api::class.java)
```

Status.kt

```kotlin
package com.example.pixabayimages.networking

enum class Status {
    SUCCESS,
```

```
    ERROR,
    LOADING
}
```

Preferences.kt

```
package com.example.pixabayimages.persistence

import android.content.Context
import android.content.SharedPreferences
import com.example.pixabayimages.model.PixabayResponse
import com.example.pixabayimages.model.UserData
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken
import java.lang.reflect.Type


class Preferences(context: Context) {
    private val preferences: SharedPreferences =
        context.getSharedPreferences("prefs", Context.MODE_PRIVATE)

    fun setFavoriteList(data: PixabayResponse.Photo?) : Preferences{
        val savepreference = getFavoriteList()
        savepreference.add(data)
        preferences.edit().putString("favoriteList",
Gson().toJson(savepreference)).apply()
        return this
    }

    fun getFavoriteList(): ArrayList<PixabayResponse.Photo?>{
        val string = preferences.getString("favoriteList", "").toString()
        string.ifEmpty { return arrayListOf() }
        val listType: Type = object :
TypeToken<ArrayList<PixabayResponse.Photo?>?>() {}.type
        return Gson().fromJson(string, listType)
    }


    fun getQuery(): String {
        return preferences.getString("query", "beach").toString()
    }

    fun setQuery(data : String?): Preferences{
        preferences.edit().putString("query", data).apply()
        return this
    }

    fun getUserList(): ArrayList<UserData?>{
        val string = preferences.getString("userList", "").toString()
        string.ifEmpty { return arrayListOf() }
        val listType: Type = object : TypeToken<ArrayList<UserData?>?>() {}.type
        return Gson().fromJson(string, listType)
    }

    fun setUserList(data: ArrayList<UserData?>) : Preferences{
        preferences.edit().putString("userList", Gson().toJson(data)).apply()
        return this
    }
```

```
    fun setCurrentUser(data : UserData) : Preferences{
        preferences.edit().putString("currentUser", Gson().toJson(data)).apply()
        return this
    }

    fun getCurrentUser() : UserData?{
        val string = preferences.getString("currentUser", "").toString()
        if (string.isEmpty()) return UserData()
        val listType: Type = object : TypeToken<UserData>() {}.type
        return Gson().fromJson(string, listType)}

}
```

BaseAplicatioin.kt

```
package com.example.pixabayimages

import android.annotation.SuppressLint
import android.app.Application
import android.content.Context
import androidx.multidex.MultiDex
import org.koin.android.ext.koin.androidContext
import org.koin.android.ext.koin.androidLogger
import org.koin.core.context.startKoin

class BaseApplication : Application() {
    companion object{
        @SuppressLint("StaticFieldLeak")
        lateinit var context : Context
    }

    override fun onCreate() {
        super.onCreate()
        startKoin {
            androidLogger()
            androidContext(this@BaseApplication)
            modules(listOf(koinModules))
        }
        context = applicationContext
    }

    override fun attachBaseContext(base: Context) {
        super.attachBaseContext(base)
        MultiDex.install(this)
    }
}
```

Constant.kt

```
package com.example.pixabayimages
const val BASE_URL = "https://pixabay.com/"
const val TAG = "asdf"
const val PER_PAGE = 20
```

Helper.kt

```kotlin
package com.example.pixabayimages

fun toMb(bytes : Int) : Float{
    return bytes / 1_000_000f
}
```

KoinModules.kt

```kotlin
package com.example.pixabayimages

import com.example.pixabayimages.fragment.FragmentDetails
import com.example.pixabayimages.fragment.FragmentAccount
import com.example.pixabayimages.fragment.FragmentExplore
import com.example.pixabayimages.fragment.FragmentFavorite
import com.example.pixabayimages.networking.*
import com.example.pixabayimages.persistence.Preferences
import org.koin.android.ext.koin.androidApplication
import org.koin.dsl.module


val koinModules = module {
    single { Repository(get(), get()) }
    single { ApiReq() }
    factory { provideOkHttpClient(get()) }
    factory { provideForecastApi(get()) }
    factory { provideLoggingInterceptor() }
    single { provideRetrofit(get()) }
    factory { ResponseHandler() }
    single { FragmentAccount() }
    single { FragmentExplore() }
    single { FragmentFavorite() }
    single { MemoryDb() }
    single { Preferences(androidApplication()) }
    single { FragmentDetails() }
}
```

LoginActivity.kt

```kotlin
package com.example.pixabayimages

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import com.example.pixabayimages.BaseApplication.Companion.context
import com.example.pixabayimages.fragment.FragmentAccount
import com.example.pixabayimages.model.UserData
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.activity_login.*
import org.koin.android.ext.android.inject
import java.util.*
import kotlin.math.log

class LoginActivity : AppCompatActivity() {
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
```

```
    private val fragmentAccount: FragmentAccount by inject()


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        btnLogin.setOnClickListener {
            userLogin()
        }

        btnRegisterInstead.setOnClickListener{
            userRegister()
        }
    }

    private fun userRegister() {
        val intent = Intent(this, RegisterActivity::class.java)
        return startActivity(intent)
    }

    private fun userLogin(): Boolean {
        val typedUsername =
username_login.text.toString().toLowerCase(Locale.ROOT)
        val typedPassword = password_login.text.toString()

        val userList = preferences.getUserList()
        var userData : UserData? = null
        for (item in userList){
            if (item?.username.equals(typedUsername)){
                userData = item

            }
        }
        val userPassword = userData?.password.toString()
        Log.d(TAG, "userLogin: yes $userPassword")
        if (typedPassword == userPassword){
            if (userData != null) {
                preferences.setCurrentUser(userData)
            }
            Log.d(TAG, "userLogin: password is same")
            val intent = Intent(this, MainActivity::class.java)
            intent.flags = Intent.FLAG_ACTIVITY_REORDER_TO_FRONT;
            fragmentAccount.setUsername(userData!!.username)
            return startActivityIfNeeded(intent, 0);
        }

        Toast.makeText(context, "Login Failed!", Toast.LENGTH_SHORT).show()

        return true
    }

}
```

MainActivity.kt

```
package com.example.pixabayimages
```

```kotlin
import android.content.Intent
import android.os.Bundle
import android.service.autofill.UserData
import androidx.appcompat.app.AppCompatActivity
import androidx.viewpager2.widget.ViewPager2
import com.example.pixabayimages.adapter.ViewPagerAdapter
import com.example.pixabayimages.fragment.FragmentAccount
import com.example.pixabayimages.fragment.FragmentExplore
import com.example.pixabayimages.fragment.FragmentFavorite
import com.example.pixabayimages.networking.ApiReq
import com.example.pixabayimages.persistence.Preferences
import kotlinx.android.synthetic.main.activity_main.*
import org.koin.android.ext.android.inject

class MainActivity : AppCompatActivity() {
    private val apiReq: ApiReq by inject()
    private val fragmentAccount: FragmentAccount by inject()
    private val fragmentExplore: FragmentExplore by inject()
    private val favoriteFragment: FragmentFavorite by inject()
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
    private var pagerAdapter: ViewPagerAdapter? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        pager.adapter = loadFragments()
        pager.registerOnPageChangeCallback(object :
ViewPager2.OnPageChangeCallback(){
            override fun onPageSelected(position: Int) {
                when(position){
                    0 -> bottomNavigationView.selectedItemId = R.id.explore
                    1 -> bottomNavigationView.selectedItemId = R.id.favorite
                    2 -> bottomNavigationView.selectedItemId = R.id.account
                }
                super.onPageSelected(position)
            }
        })

        bottomNavigationView.setOnNavigationItemSelectedListener { it ->
            when (it.itemId){
                R.id.explore -> pager.setCurrentItem(0, true)
                R.id.favorite -> pager.setCurrentItem(1, true)
                R.id.account -> pager.setCurrentItem(2, true)
            }
            true
        }
        memoryDb.currentUser.observe(this,{
            preferences.setCurrentUser(it)
            if(it.username == ""){
                doLogin()
            }
        })
        doLogin()
    }

    private fun doLogin() {
        val currentUser = preferences.getCurrentUser()
        if (currentUser?.username == ""){
```

```
            val intent = Intent(this, LoginActivity::class.java)
            intent.flags = Intent.FLAG_ACTIVITY_REORDER_TO_FRONT;
            startActivityIfNeeded(intent, 0);
        }
    }


    private fun loadFragments(): ViewPagerAdapter {
        val viewPagerAdapter = ViewPagerAdapter(this)
        viewPagerAdapter.replaceAll(
                arrayListOf(fragmentExplore, favoriteFragment, fragmentAccount)
        )
        return viewPagerAdapter
    }



//    fun reload(v: View?){
//        apiReq.ping((Math.random()*100).toInt()).observe(this, {
//            status.text = it.status.toString()
//            if (it.status == Status.SUCCESS) {
//                status.text = ""
//                Log.d(TAG, it.response?.hits.toString())
//                val url = it.response?.hits?.get(0)?.webformatURL
//                Glide.with(context).load(url).centerCrop()
//                        .into(image)
//            }
//        })
//    }
}
```

MemoryDB.kt

```
package com.example.pixabayimages

import android.graphics.drawable.Drawable
import androidx.lifecycle.MutableLiveData
import com.example.pixabayimages.model.ImageDetails
import com.example.pixabayimages.model.UserData
import com.example.pixabayimages.model.PixabayResponse.*
import org.koin.core.KoinComponent

class MemoryDb : KoinComponent {
    val imageData: MutableLiveData<Photo> by lazy {
        MutableLiveData<Photo>()
    }

    val previewImage: MutableLiveData<Drawable> by lazy {
        MutableLiveData<Drawable>()
    }
    val query: MutableLiveData<String> by lazy {
        MutableLiveData<String>()
    }

    val imageDetails: MutableLiveData<ImageDetails> by lazy {
        MutableLiveData<ImageDetails>()
    }
```

```
    val placeHolderImg: MutableLiveData<Drawable> by lazy {
        MutableLiveData<Drawable>()
    }

    val currentUser: MutableLiveData<UserData> by lazy {
        MutableLiveData<UserData>()
    }


}
```

RegisterActivity.kt

```
package com.example.pixabayimages

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import com.example.pixabayimages.model.UserData
import com.example.pixabayimages.persistence.Preferences

import kotlinx.android.synthetic.main.activity_register.*
import org.koin.android.ext.android.inject
import java.util.*

class RegisterActivity : AppCompatActivity() {
    private val memoryDb: MemoryDb by inject()
    private val preferences: Preferences by inject()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)


        btnRegister.setOnClickListener{
            userRegister()
        }
        btnLoginInstead.setOnClickListener{
            userLogin()
        }

    }

    private fun userRegister(){
        val userList = preferences.getUserList()

        val typedUsername = username_register.text.toString().toLowerCase()
        val typedPassword = password_register.text.toString()
        val typedConfirmationPassword = confirmation_register.text.toString()

        if(typedPassword != typedConfirmationPassword){
            Toast.makeText(BaseApplication.context, "Password not same!!",
Toast.LENGTH_SHORT).show()
            return
        }
        val newUser = UserData()
```

```
        newUser.username = typedUsername
        newUser.password = typedPassword

        userList.add(newUser)
        preferences.setUserList(userList)
        Toast.makeText(BaseApplication.context, "Registered!",
Toast.LENGTH_SHORT).show()
        userLogin()


    }

    private fun userLogin() {
        val intent = Intent(this, LoginActivity::class.java)
        return startActivity(intent)
    }
}
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/pixabay_gallery"
        android:textSize="40sp"
        android:layout_above="@id/textFields"
        android:layout_marginBottom="120dp"
        android:fontFamily="@font/seoul_hangang"/>
    <TextView
        android:layout_width="200dp"
        android:layout_height="20dp"
        android:text="@string/textview"
        android:layout_alignStart="@id/textFields"
        android:layout_marginBottom="16dp"
        android:layout_above="@id/textFields"
        android:layout_centerHorizontal="true"
        android:fontFamily="sans-serif-light"/>
    <RelativeLayout
        android:id="@+id/textFields"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true">

        <EditText
            android:id="@+id/username_login"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/username_login"
            android:inputType="textPersonName"
```

```
            android:drawableStart="@drawable/ic_person_16"
            android:drawablePadding="16dp"
            android:background="@drawable/shape_edittext_round_backround"
            android:paddingVertical="8dp"
            android:paddingStart="24dp"
            android:paddingEnd="24dp"
            android:drawableTint="@color/grey_500"/>

        <EditText
            android:paddingVertical="8dp"
            android:paddingStart="24dp"
            android:paddingEnd="24dp"
            android:id="@+id/password_login"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_below="@id/username_login"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:ems="10"
            android:hint="@string/password_login"
            android:inputType="textPassword"
            tools:ignore="TextFields"
            android:background="@drawable/shape_edittext_round_backround"
            android:drawableStart="@drawable/ic_logout_16"
            android:drawablePadding="16dp"
            android:drawableTint="@color/grey_500"/>

    </RelativeLayout>

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="104dp"
        android:layout_height="32dp"
        android:background="@drawable/shape_btn_round_background"
        android:drawableTint="@color/white"
        android:text="@string/btn_login"
        android:textSize="14sp"
        android:textColor="#FBFAFA"
        android:drawablePadding="-4dp"
        android:textAllCaps="false"
        android:textStyle="normal"
        android:layout_marginTop="32dp"
        android:layout_below="@id/textFields"
        android:layout_centerHorizontal="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/btnRegisterInstead"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/btnLogin"
        android:text="@string/register"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:textStyle="bold"
        android:background="@drawable/dashed_underline"/>
```

```
</RelativeLayout>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:elevation="20dp"
        android:translationZ="6dp"
        app:cardBackgroundColor="@color/white"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginStart="16dp"
            android:fontFamily="@font/seoul_hangang"
            android:gravity="center_vertical"
            android:text="@string/pixabay_gallery"
            android:textSize="20sp" />
    </androidx.cardview.widget.CardView>

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="40dp"
        android:layout_marginBottom="40dp"
        tools:layout_editor_absoluteY="40dp" />


    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottomNavigationView"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:background="@color/white"
        android:elevation="6dp"
        android:translationZ="20dp"
        app:itemIconSize="20dp"
        app:itemIconTint="@drawable/color_selector"
        app:itemTextColor="@drawable/color_selector"
        app:labelVisibilityMode="unlabeled"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/pager"
        app:menu="@menu/bottom_menu" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/pixabay_gallery"
        android:textSize="40sp"
        android:layout_above="@id/textFields"
        android:layout_marginBottom="120dp"
        android:fontFamily="@font/seoul_hangang"/>
    <TextView
        android:layout_width="200dp"
        android:layout_height="20dp"
        android:text="@string/create_account"
        android:layout_alignStart="@id/textFields"
        android:layout_marginBottom="16dp"
        android:layout_above="@id/textFields"
        android:layout_centerHorizontal="true"
        android:fontFamily="sans-serif-light"/>
    <RelativeLayout
        android:id="@+id/textFields"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true">

        <EditText
            android:id="@+id/username_register"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/username_login"
            android:inputType="textPersonName"
            android:drawableStart="@drawable/ic_person_16"
            android:drawablePadding="16dp"
            android:background="@drawable/shape_edittext_round_backround"
            android:paddingVertical="8dp"
            android:paddingStart="24dp"
            android:paddingEnd="24dp"
            android:drawableTint="@color/grey_500"/>

        <EditText
            android:paddingVertical="8dp"
            android:paddingStart="24dp"
            android:paddingEnd="24dp"
            android:id="@+id/password_register"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
```

```
            android:layout_below="@id/username_register"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:ems="10"
            android:hint="@string/password_login"
            android:inputType="textPassword"
            tools:ignore="TextFields"

            android:background="@drawable/shape_edittext_round_backround"
            android:drawableStart="@drawable/ic_logout_16"
            android:drawablePadding="16dp"
            android:drawableTint="@color/grey_500"/>

        <EditText
            android:paddingVertical="8dp"
            android:paddingStart="24dp"
            android:paddingEnd="24dp"
            android:id="@+id/confirmation_register"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"
            android:layout_below="@id/password_register"
            android:ems="10"
            android:hint="@string/confirm_password"
            android:inputType="textPassword"
            tools:ignore="TextFields"
            android:background="@drawable/shape_edittext_round_backround"
            android:drawableStart="@drawable/ic_logout_16"
            android:drawablePadding="16dp"
            android:drawableTint="@color/grey_500"/>
    </RelativeLayout>

    <Button
        android:id="@+id/btnRegister"
        android:layout_width="104dp"
        android:layout_height="32dp"
        android:background="@drawable/shape_btn_round_background"
        android:drawableTint="@color/white"
        android:text="@string/btn_register"
        android:textSize="14sp"
        android:textColor="#FBFAFA"
        android:drawablePadding="-4dp"
        android:textAllCaps="false"
        android:textStyle="normal"
        android:layout_marginTop="32dp"
        android:layout_below="@id/textFields"
        android:layout_centerHorizontal="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/btnLoginInstead"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/btnRegister"
        android:text="@string/login_instead"
```

```
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:textStyle="bold"
        android:background="@drawable/dashed_underline"/>



</RelativeLayout>
```

Fragment_account.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".fragment.FragmentAccount">

    <!-- TODO: Update blank fragment layout -->


    <Button
        android:id="@+id/btn_logout"
        android:layout_width="104dp"
        android:layout_height="32dp"
        android:background="@drawable/shape_btn_round_background"
        android:drawableStart="@drawable/ic_logout_16"
        android:drawableTint="@color/white"
        android:text="@string/btn_logout"
        android:textSize="14sp"
        android:textColor="#FBFAFA"
        android:drawablePadding="-4dp"
        android:paddingStart="16dp"
        android:paddingEnd="0dp"
        android:textAllCaps="false"
        android:textStyle="normal"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="16dp"
        android:layout_alignParentEnd="true"/>

    <TextView
        android:id="@+id/userName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/pfpCard"
        android:layout_centerHorizontal="true"
        android:textSize="48sp"
        android:text="@string/account_name"
        android:textColor="@color/black"
        android:layout_marginTop="24dp"/>

    <androidx.cardview.widget.CardView
        android:id="@+id/pfpCard"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_centerHorizontal="true"
        app:cardCornerRadius="250dp"
```

```
        android:layout_centerVertical="true">

        <ImageView
            android:id="@+id/pfp"
            android:layout_width="250dp"
            android:layout_height="250dp"
            android:src="@drawable/mark" />
    </androidx.cardview.widget.CardView>

</RelativeLayout>
```

Fragment_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_height="wrap_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:nestedScrollingEnabled="true">
    <RelativeLayout xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        tools:context=".fragment.FragmentDetails">

        <androidx.cardview.widget.CardView
            android:id="@+id/pfpCard"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:cardCornerRadius="24dp"
            android:elevation="0dp"
            android:translationZ="-10dp"
            android:layout_marginStart="16dp"
            android:layout_marginVertical="8dp">

            <ImageView
                android:id="@+id/pfp"
                android:layout_width="32dp"
                android:layout_height="32dp"
                android:src="@drawable/ic_person_foreground"
                android:contentDescription="@string/user_pfp" />
        </androidx.cardview.widget.CardView>

        <TextView
            android:id="@+id/userName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_toEndOf="@id/pfpCard"
            android:text="@string/username"
            android:textColor="@color/black"
            android:fontFamily="sans-serif"
            android:layout_alignTop="@id/pfpCard"
            android:layout_alignBottom="@id/pfpCard"
            android:gravity="center_vertical"
            android:layout_marginStart="8dp"/>
        <ImageView
            android:id="@+id/largeImage"
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/pfpCard"
            android:adjustViewBounds="true"
            android:src="@drawable/ic_image_foreground"
            android:contentDescription="@string/large_image" />
        <ImageView
            android:id="@+id/favorite"
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:layout_below="@id/largeImage"
            android:src="@drawable/ic_star_favorite_foreground"
            android:layout_marginHorizontal="16dp"
            android:layout_marginVertical="8dp"
            android:contentDescription="@string/favorite_icon" />
        <ImageView
            android:id="@+id/download"
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:layout_below="@id/largeImage"
            android:layout_toEndOf="@+id/favorite"
            android:layout_alignTop="@id/favorite"
            android:layout_marginHorizontal="16dp"
            android:src="@drawable/ic_download_foreground"
            android:contentDescription="@string/download_icon" />
        <ImageView
            android:id="@+id/setAs"
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:layout_below="@id/largeImage"
            android:layout_toEndOf="@id/download"
            android:layout_alignTop="@id/favorite"
            android:layout_marginHorizontal="16dp"
            android:src="@drawable/ic_set_as_foreground"
            android:contentDescription="@string/set_as_icon" />
        <TextView
            android:textSize="14sp"
            android:layout_marginEnd="3dp"
            android:id="@+id/likes"
            android:layout_below="@id/favorite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/empty"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_marginStart="16dp"/>
        <TextView
            android:textSize="14sp"
            android:layout_toEndOf="@id/likes"
            android:layout_below="@id/favorite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/likes"
            android:textStyle="bold"
            android:textColor="@color/black"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
```

```
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/likes"
            android:text="@string/tags"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/tags"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/likes"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/tags"
            android:text="@string/resolution"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/resolution"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/tags"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/resolution"
            android:text="@string/size"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/size"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
```

```
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/resolution"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/size"
            android:text="@string/views"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/views"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/size"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/views"
            android:text="@string/downloads"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/downloads"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/views"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/downloads"
```

```
            android:text="@string/comments"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <TextView
            android:id="@+id/comments"
            android:layout_marginEnd="16dp"
            android:layout_alignParentEnd="true"
            android:textColor="@color/black"
            android:textSize="12sp"
            android:fontFamily="sans-serif-light"
            android:layout_marginTop="8dp"
            android:layout_marginStart="16dp"
            android:layout_below="@id/downloads"
            android:text="@string/empty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"/>


    </RelativeLayout>
</ScrollView>
```

Fragment_explorer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.appcompat.widget.SearchView
        android:id="@+id/searchView"
        android:layout_width="match_parent"
        android:layout_height="32dp"
        android:layout_marginTop="20dp"
        android:layout_marginHorizontal="40dp"
        app:queryHint="Search"
        app:queryBackground="@android:color/transparent"
        android:queryHint="hint"
        app:defaultQueryHint="hint"
        android:iconifiedByDefault="false"
        app:iconifiedByDefault="false"
        android:background="@drawable/shape_edittext_round_backround"/>

    <LinearLayout
        android:id="@+id/noResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/searchView"
        android:orientation="horizontal"
        android:gravity="center_horizontal"
        android:visibility="visible"
        android:layout_marginVertical="6dp"
        android:layout_below="@id/searchView">
        <TextView
            android:id="@+id/noResultsFor"
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
            android:text="@string/no_results_for"
            android:gravity="center"
            android:textSize="20sp"
            android:visibility="gone"
            android:fontFamily="sans-serif-light" />
        <TextView
            android:id="@+id/resultsFor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/results_for"
            android:gravity="center"
            android:textSize="20sp"
            android:fontFamily="sans-serif-light" />
        <TextView
            android:id="@+id/query"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/this_query"
            android:layout_marginStart="3dp"
            android:gravity="center"
            android:textSize="20sp"
            android:fontFamily="sans-serif-medium" />

    </LinearLayout>


    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/refresh"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/noResult"
        android:layout_marginTop="-6dp">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/list"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="top"
            android:visibility="visible"
            app:layoutManager="androidx.recyclerview.widget.GridLayoutManager"

            app:layout_constraintTop_toBottomOf="@id/noResult"
            app:spanCount="3"
            tools:itemCount="15"
            tools:listitem="@layout/item_explore" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>


</RelativeLayout>
```

Fragment_favorite.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".fragment.FragmentFavorite">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/listfavourite"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:visibility="visible"
        app:layoutManager="androidx.recyclerview.widget.GridLayoutManager"

        app:layout_constraintTop_toBottomOf="@id/noResult"
        app:spanCount="3"
        tools:itemCount="15"
        tools:listitem="@layout/item_explore" />

</FrameLayout>
```

Item_explorer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="4dp"
    android:background="?android:attr/selectableItemBackground"
    android:layout_height="wrap_content">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="4dp"
        android:elevation="0dp"
        android:translationZ="-10dp">

        <com.example.pixabayimages.customView.SquareImageView
            android:id="@+id/image"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_image_foreground"
            android:background="@color/grey_100"/>
    </androidx.cardview.widget.CardView>
</RelativeLayout>
```

Kontribusi Tim

- Faruq : Membuat Halaman Akun dan Login, serta laporan bagian gambar implementasi, tata cara penggunaan, dan bagian penutup
- Shofi : Membuat halaman Favourite, serta laporan bagian enjelasan fungsionalitas aplikasi, menu, dan pembuatan (code)Aplikasi)

- Mirza : Melakukan setup API dan klien API, Membuat halaman FragmentExplore (layout dan kode), Membuat halaman Register (Layout dan Kode)

Penutup

Begitulah hasil aplikasi kami yang berjudul Pixabay Gallery, mohon maaf bila ada penyampaian yang kurang jelas atau kurang bisa dimengerti. Lebih dan kurangnya kami mohon maaf. Kami harap untuk bapak dosen yang mengampu matakuliah Aplikasi Komputasi Bergerak bisa memberikan nilai yang maksimal untuk projek yang kami kerjakan ini.

Daftar Pustaka:

1. "Pixabay home page". Pixabay.com. Archived from the original on 2019-01-30. Retrieved 2018-10
2. *"Terms of Service". web.archive.org. 2019-01-08. Retrieved 2020-11-14.*
3. *"Pixabay - Terms of Service". Retrieved 2020-11-14.*