BSc (Hons) in Information Technology

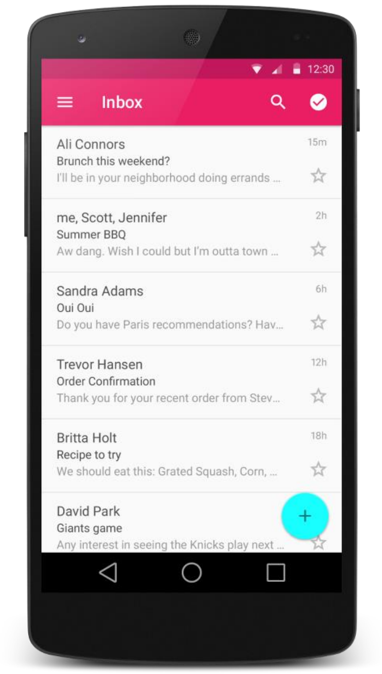
Year 2

IT2010 - Mobile Application Development

Lab sheet 4

Android - RecyclerView

If your app needs to display a scrolling list of elements based on large data sets (or data that frequently changes), you should use RecyclerView.



Ali Connors 15m
Brunch this weekend?
I'll be in your neighborhood doing errands ...

me, Scott, Jennifer 2h
Summer BBQ
Aw dang. Wish I could but I'm outta town ...

Sandra Adams 6h
Oui Oui
Do you have Paris recommendations? Hav...

Trevor Hansen 12h
Order Confirmation
Thank you for your recent order from Stev...

Britta Holt 18h
Recipe to try
We should eat this: Grated Squash, Corn, ...

David Park
Giants game
Any interest in seeing the Knicks play next ...

Read more on RecyclerView from Android Developers website.
https://developer.android.com/guide/topics/ui/layout/recyclerview

In this tutorial let's use some interesting libraries to make your applications beautiful. Apart from RecyclerView, you will learn how to import images directly from the internet to your application. Therefore, you can reduce the app size. The next thing you will learn to use is Circular image view. It's true that you can make an image circular by coding but this way it's easier and faster.

Add the following dependencies in the Gradle app file.

1. RecyclerView

```
//recyclerview
implementation 'com.android.support:recyclerview-v7:28.0.0'
```

Note that you must change the compileSdkVersion, tagetSdkVersion buildToolsVersion according to the version of the RecyclerView version. In this case '28.0.0'.

2. Glide: Library to use internet images

```
//glide
implementation 'com.github.bumptech.glide:glide:4.11.0'
annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'
```

3. Circular Image view

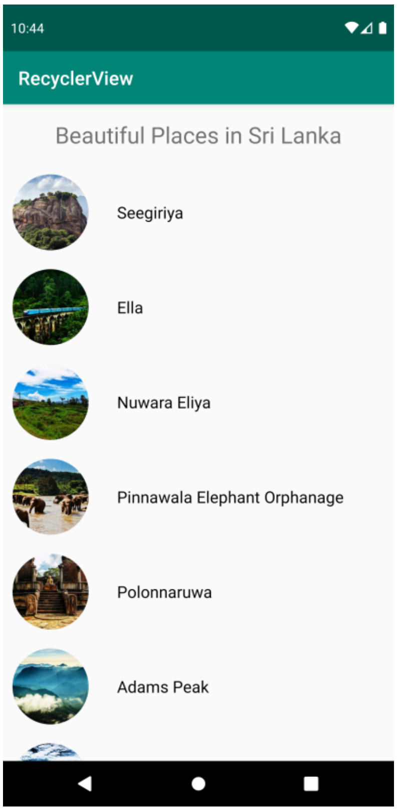
```
//circle imageview
implementation 'de.hdodenhof:circleimageview:3.1.0'
```

If your app needs to connect to the internet, you must give the app permission to do so. Set the permission as follows in the ‘AndroidManifest.xml’ file.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Now everything is ready to implement the code for the RecycleView.

We are going to develop the following application



Beautiful Places in Sri Lanka

Seegiriya

Ella

Nuwara Eliya

Pinnawala Elephant Orphanage

Polonnaruwa

Adams Peak

First, let’s look at layouts. Initially we should create an additional layout to display the Recycler view as a scrollable list view. That layout can be inflated into the RecyclerView in the activity layout through the code. Then we can load any amount of information in to the RecyclerView.

You can load data from an internal file, SQLite database, Firebase or any other API into your app. Let your creativity decide how it should be displayed.

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">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:text="@string/title"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        tools:layout_editor_absoluteX="0dp">
    </androidx.recyclerview.widget.RecyclerView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

layout_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:id="@+id/parent_layout"
    android:padding="10dp">
    <de.hdodenhof.circleimageview.CircleImageView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:id="@+id/image"
        android:src="@mipmap/ic_launcher_round"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Canada"
        android:id="@+id/image_name"
        android:layout_toRightOf="@id/image"
        android:textColor="#000"
        android:layout_centerVertical="true"
        android:layout_marginLeft="30dp"
        android:textSize="17sp"
        />
</RelativeLayout>
```

That’s all from the layout side. Magic happens in the code. First you must create an adapter class to handle data and display as you want. In the activity class you can create an object from the adapter class and get the job done. Now let’s look at the classes.

ReclerViewAdapter.java

```
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
    private static final String TAG = "test.sliit.recyclerview.RecyclerViewAdapter";
    private ArrayList<String> mImageNames = new ArrayList<>();
    private ArrayList<String> mImage = new ArrayList<>();
    private Context mContext;
    public RecyclerViewAdapter(ArrayList<String> mImageNames, ArrayList<String>
mImage, Context mContext) {
        this.mImageNames = mImageNames;
        this.mImage = mImage;
        this.mContext = mContext;
    }
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.layout_list,parent,false);
        ViewHolder viewHolder = new ViewHolder(view);

        return viewHolder;
    }
    @SuppressLint("LongLogTag")
    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, final int position) {
        Log.d(TAG, "onBindViewHolder: called");
        Glide.with(mContext)
            .asBitmap().load(mImage.get(position))
            .into(holder.image);

        holder.imageName.setText(mImageNames.get(position));
        holder.parentLayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Log.d(TAG, "onClick: clicked on"+mImageNames.get(position));


Toast.makeText(mContext,mImageNames.get(position),Toast.LENGTH_SHORT).show();
            }
        });
    }
    @Override
    public int getItemCount() {
        return mImageNames.size();
    }
    public class ViewHolder extends RecyclerView.ViewHolder{
        CircleImageView image;
        TextView imageName;
        RelativeLayout parentLayout;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            image = itemView.findViewById(R.id.image);
            imageName = itemView.findViewById(R.id.image_name);
            parentLayout = itemView.findViewById(R.id.parent_layout);
        }
    }
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private static final String TAG = "MainActivity";
    //vars
    private ArrayList<String> mNames = new ArrayList<>();
    private ArrayList<String> mImageUrls = new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate: started");
        initImageBitmaps();
    }
    private void initImageBitmaps(){
        Log.d(TAG, "initImageBitmaps: started");
        mImageUrls.add("https://lp-cms-production.imgix.net/2019-
06/b4fbc706dab2a70a96588309ed268a1a-sri-lanka.jpeg");
        mNames.add("Seegiriya");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Demodara-Nine-Arch-Bridge.jpg");
        mNames.add("Ella");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Train-ride-from-Kandy-to-Nuwara-Eliya.jpg");
        mNames.add("Nuwara Eliya");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Pinnawala-Elephant-Orphanage.jpg");
        mNames.add("Pinnawala Elephant Orphanage");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Ruins-of-Polonnaruwa.jpg");
        mNames.add("Polonnaruwa");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Adams-Peak.jpg");
        mNames.add("Adams Peak");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Mirissa-Fisheries-Harbor.jpg");
        mNames.add("Mirissa");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Leopards.jpg");
        mNames.add("Yala National Park");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Colombo.jpg");
        mNames.add("Colombo");
        mImageUrls.add("https://img.traveltriangle.com/blog/wp-content/tr:w-700,h-
400/uploads/2015/06/Jaffna.jpg");
        mNames.add("Jaffna");
        initRecyclerView();
    }
    private  void initRecyclerView(){
        Log.d(TAG, "initRecyclerView: started");
        RecyclerView recyclerView = findViewById(R.id.recycler_view);
        RecyclerViewAdapter adapter = new RecyclerViewAdapter(mNames,mImageUrls,this);
        recyclerView.setAdapter(adapter);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
    }
}
```

Refer following for more:

1. https://developer.android.com/guide/topics/ui/layout/recyclerview
2. https://www.youtube.com/watch?v=Vyqz_-sJGFk
3. http://www.codeplayon.com/2018/10/changing-background-color-of-selected-item-in-recyclerview/