Week 5 Workshop Activity

Preferences

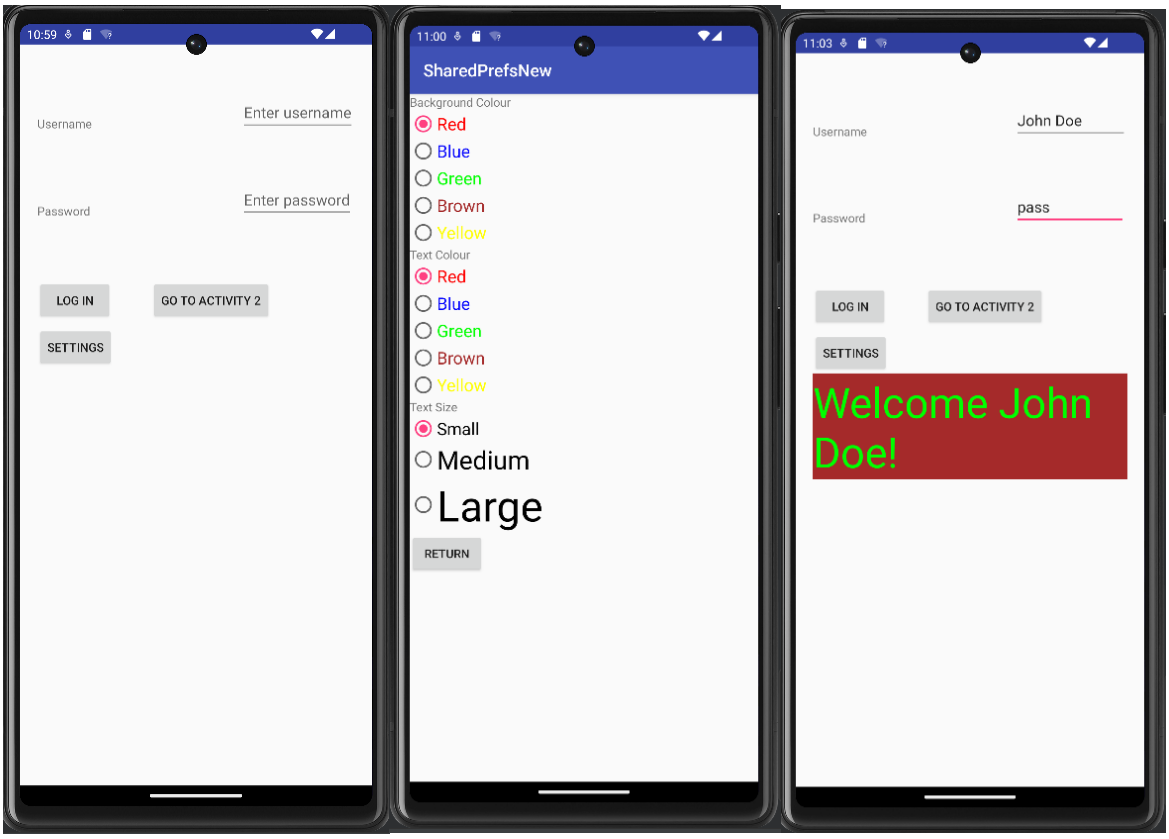
Objectives of Workshop 5:

1. Shared preferences
2. Build a UI containing a recycler view

1. Adding User Preferences

Start by downloading the SharedPrefsNew-incomplete-fall2022.zip file on Canvas. By default, this file has a log-in page that stores log-in information to shared preferences. This information can be retrieved and displayed in Activity2.

The goal is to add a settings activity into this application. This activity lets the user set the **background color, text color and text size.** These settings should be saved as shared preferences. When the user logs in, a “Welcome [username]” message should be displayed with text having the background color, text color and size as set in preferences.



Implement this on your own as much as possible. As a hint, your submit function in **MainActivity** should look like this:

```
public void submit (View view){
    SharedPreferences sharedPrefs = getSharedPreferences( name: "MyData", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPrefs.edit();
    String username = sharedPrefs.getString( s: "username", DEFAULT);
    String password = sharedPrefs.getString( s: "password", DEFAULT);
    String color = sharedPrefs.getString( s: "colour", DEFAULT);
    String textcolor = sharedPrefs.getString( s: "textcolour", DEFAULT);
    String textsize = sharedPrefs.getString( s: "textsize", DEFAULT);

    if (username.equals(usernameEditText.getText().toString())||password.equals(passwordEditText.getText().toString()))
    {
        welcomeText.setText("Welcome "+username+"!");

        if(textcolor != DEFAULT && textcolor != "") {
            welcomeText.setTextColor(Color.parseColor(textcolor));
        }
        if(color != DEFAULT && textcolor != "") {
            welcomeText.setBackgroundColor(Color.parseColor(color));
        }
        if(textsize != DEFAULT && textsize != "") {
            welcomeText.setTextSize(TypedValue.COMPLEX_UNIT_SP, Float.parseFloat(textsize));
        }
    }
    else {
        Toast.makeText( context: this,  text: "Username and password saved to Preferences", Toast.LENGTH_LONG).show();

        editor.putString( s: "username", usernameEditText.getText().toString());
        editor.putString( s: "password", passwordEditText.getText().toString());
        editor.commit();
    }
}
```

2. Recycler View

Start by downloading the RecyclerViewNew-fall2022.zip file on Canvas. This application uses a RecyclerView to display a list of items as a scrollable list and associate click behaviors with the list items.

Change the data for the RecyclerView to be a list of 20 similar items of your choice – e.g., names of people, geographical locations, universities, etc. You must find your own set of 20 images related to these items, and add them to the drawables folder.

Each item should be displayed in a row, showing the name of the item in a TextView or EditText and an image in an ImageView.

When the user taps an item in a row, a Toast should appear with the name of the item clicked, and also a new activity should open with detailed information corresponding to the clicked item.

To do this, create a new Java class called ListItem:

```java
package iat359course.ca.recyclerviewnew;

import java.util.List;

public class ListItem {
    public String name;
    public int imageID;
    public String additionalData;

    public ListItem(String name, int imageID, String additionalData){
        this.name = name;
        this.imageID = imageID;
        this.additionalData = additionalData;
    }
}
```

In your MainActivity, populate the ListItem with your own data **(replace the course names and image IDs with your own):**

```java
public class MainActivity extends Activity {
    RecyclerView myRecycler;
    RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager mLayoutManager;
    ArrayList<ListItem> courses = new ArrayList<~>();
    ArrayList<String> coursenames = new ArrayList<~>(Arrays.asList("IAT381", "IAT351","IAT336", "IAT337", "IAT338", "IAT201", "IAT401",
            "IAT111", "IAT222", "IAT333", "IAT444", "IAT555", "IAT666", "IAT111", "IAT228", "IAT999", "IAT000", "IAT339", "IAT449", "IAT559"));
    ArrayList<String> coursedata = new ArrayList<~>(Arrays.asList("About IAT381", "About IAT351","About IAT336", "About IAT337",
            "About IAT338", "About IAT201", "About IAT401", "About IAT111", "About IAT222", "About IAT333", "About IAT444", "About IAT555",
            "About IAT666", "About IAT111", "About IAT228", "About IAT999", "About IAT000", "About IAT339", "About IAT449", "About IAT559"));
    ArrayList<Integer> images;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myRecycler = (RecyclerView) findViewById(R.id.my_recycler_view);

        // use a linear layout manager
        mLayoutManager = new LinearLayoutManager( context: this);
        myRecycler.setLayoutManager(mLayoutManager);

        images = new ArrayList<Integer>(Arrays.asList(R.drawable.i1, R.drawable.i2, R.drawable.i3, R.drawable.i4, R.drawable.i5,
                R.drawable.i6, R.drawable.i7, R.drawable.i8, R.drawable.i9, R.drawable.i10, R.drawable.i11, R.drawable.i12, R.drawable.i13,
                R.drawable.i14, R.drawable.i15, R.drawable.i16, R.drawable.i17, R.drawable.i18, R.drawable.i19, R.drawable.i20));

        for(int i = 0; i<20; i++){
            courses.add(new ListItem(coursenames.get(i), images.get(i), coursedata.get(i) ));
        }

        adapter = new MyAdapter(courses, getApplicationContext());
        myRecycler.setAdapter(adapter);
    }
}
```

Edit the MyViewHolder class in MyAdapter.java so that it can display images.

```
public static class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {

    public TextView myTextView;
    public ImageView myImageView;
    public String data;
    Context context;

    public MyViewHolder(View itemView) {
        super(itemView);
        myTextView = (TextView) itemView.findViewById(R.id.textview);
        myImageView =  (ImageView) itemView.findViewById(R.id.imageview);
        itemView.setOnClickListener(this);
        context = itemView.getContext();

    }
```

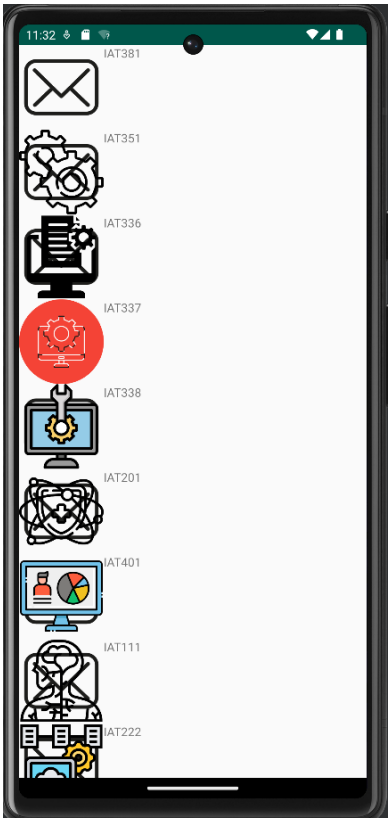
Change MyAdapter to display images:

```
@Override
public MyAdapter.MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.listrow, parent, attachToRoot: false);
    MyViewHolder viewHolder = new MyViewHolder(v);
    return viewHolder;
}

@Override
public void onBindViewHolder(MyAdapter.MyViewHolder holder, int position) {
    //TextView tv = (TextView) holder.itemView;
    TextView tv = holder.myTextView;
    ImageView iv = holder.myImageView;
    holder.data = list.get(position).additionalData;
    tv.setText(list.get(position).name.toString());
    iv.setImageResource(list.get(position).imageID);
}
```

Note: R.layout.listrow in the image above refers to an XML file that you should create from scratch. This should be a LinearLayout containing an ImageView and a TextView.

The end result should look something like this:



Change the onClick() function under MyAdapter so that clicking on an item leads to an Activity like this: