Fit-A-Roo: A Fitness App

The way this app flow is that the main screen is the welcome screen, once the user presses the button they will go to the next activity which will allow them two either long-click to view details about the activity such as how to perform it and what body part it focuses or a click to select the item in the list to use for this workout. Once the user has selected the exercises, they will be able to move on to the next screen where they can track the number of sets and reps for each activity.

Phase One: Create MainActivity, Exercise class, and Workout class

1) Create an empty activity for the MainActivity which will serve as a welcome screen.
2) Add a button that moves us to next activity.

Create a class in the folder with the MainActivity called Exercise.

This class holds data regarding a particular exercise which includes the name of the exercise, the body part it focuses on, and how to perform this exercise.

1) Exercise class has 3 private strings for exerciseName, bodyPart, and howTo, then create a private Integer ArrayList to hold the number of reps for this particular exercise, lastly create 2 private integers named sets and selectedSets. This class implements cloneable.
2) Include two constructors, one default and one copy constructors.
   public Exercise(String exerciseName, String bodyPart, String howTo)
   public Exercise(Exercise exercise)
3) Provide appropriate accessors and mutators for private fields.
4) Next implement clone()
5) Provide a way to add to reps arraylist as well as increment or decrement value of rep for a particular set.

Workout class holds data regarding exercises that the user has selected to perform.

1) This class has a private ArrayList<Exercise> selectedExercises field.
2) A Default constructor that initializes selectedExercises
   Public Workout()
3) Provide a way to add and remove from arraylist and appropriate accessors/mutators.

Phase 3: SelectWorkoutActivity

This phase will involve implementing the RecyclerView, a custom adapter, and an item click listener for recyclerview to allow for adding selected items to a list and a long click to show details about the clicked exercise.

1) Create another empty activity class SelectWorkOutActivity, add a recyclerview and a button to this activity.
2) Create a new class for our custom adapter that will adapt a provide array list to display items of the ArrayList in the recyclerview. The contents for this is provided on the developer.android.com but you will need to adapt it to your program.

```
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;


import java.util.ArrayList;

public class CustomAdapter extends
RecyclerView.Adapter<RecyclerView.ViewHolder> {

   private final Context context;
   private ArrayList<Exercise> list;

   public CustomAdapter(Context context, ArrayList<Exercise> list) {
      this.context = context;
      this.list = list;
   }

   private class ViewHolder extends RecyclerView.ViewHolder {

      TextView yourView;
      ViewHolder(final View itemView) {
         super(itemView);
         yourView = itemView.findViewById(R.id.textHolder); //
         // Initialize your All views present in list items
```

```
        }
        void bind(int position) {
            // This method will be called anytime a list item is
created or update its data
            //Do your stuff here
            yourView.setText(list.get(position).toString());
        }
    }

    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        return new
ViewHolder(LayoutInflater.from(context).inflate(R.layout.text_row_it
em,
                parent, false));
    }

    @Override
    public void onBindViewHolder(RecyclerView.ViewHolder holder, int
position) {
        ((ViewHolder) holder).bind(position);
    }

    @Override
    public int getItemCount() {
        return list.size();
    }
}
```

3) Next create another class RecyclerItemClickListener which will hold the following content because Recycler View does not support item click listening natively.

```
import android.content.Context;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import androidx.recyclerview.widget.RecyclerView;

public class RecyclerItemClickListener implements
RecyclerView.OnItemTouchListener {
    private OnItemClickListener mListener;

    public interface OnItemClickListener {
        public void onItemClick(View view, int position);
```

```
      public void onLongItemClick(View view, int position);
   }

   GestureDetector mGestureDetector;

   public RecyclerItemClickListener(Context context, final RecyclerView
recyclerView, OnItemClickListener listener) {
      mListener = listener;
      mGestureDetector = new GestureDetector(context, new
GestureDetector.SimpleOnGestureListener() {
         @Override
         public boolean onSingleTapUp(MotionEvent e) {
            return true;
         }

         @Override
         public void onLongPress(MotionEvent e) {
            View child = recyclerView.findChildViewUnder(e.getX(),
e.getY());
            if (child != null && mListener != null) {
               mListener.onLongItemClick(child,
recyclerView.getChildAdapterPosition(child));
            }
         }
      });
   }

   @Override public boolean onInterceptTouchEvent(RecyclerView view,
MotionEvent e) {
      View childView = view.findChildViewUnder(e.getX(), e.getY());
      if (childView != null && mListener != null &&
mGestureDetector.onTouchEvent(e)) {
         mListener.onItemClick(childView,
view.getChildAdapterPosition(childView));
         return true;
      }
      return false;
   }

   @Override public void onTouchEvent(RecyclerView view, MotionEvent
motionEvent) { }

   @Override
   public void onRequestDisallowInterceptTouchEvent (boolean
disallowIntercept){}
}
```

4) In the selectWorkOutActivity initialize the Recyclerview, custom adapter, addOnItemTouchListener, and all exercise related data.
5) Implement long click to start a new activity that shows details about the item long clicked
6) Implement the button that starts the workout with select exercises.

How to Test:

Please use the following settings for AVD:
Device: Pixel 5
Resolution: 1080 x 2340: 440dpi
API Level: Android 11.0(Google APIs)
Release Name: R
Architecture: CPU/ABI x86

Testing:

Start app.

Then press button.

Then select some exercises

Or try long clicking on some exercises to access details about activities.