

Mätverktygets kod

BaseActivity:

```
package com.example.digitalgoniometer;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.example.digitalgoniometer.model.User;
import com.google.gson.Gson;

public class BaseActivity extends AppCompatActivity {

    private Button startMeasureButton, showHistoryButton, userButton,
    backButton, infoButton;
    protected User user;
    protected SharedPreferences sharedPreferences;
    protected SharedPreferences.Editor editor;
    protected Gson gson;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_base);

        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }

        sharedPreferences =
        BaseActivity.this.getSharedPreferences("kth.majwah.sharedpreferences.size",
        BaseActivity.this.MODE_PRIVATE);
        editor = sharedPreferences.edit();
        gson = new Gson();

        startMeasureButton = findViewById(R.id.measureButton);
        showHistoryButton = findViewById(R.id.historyButton);
        userButton = findViewById(R.id.UserButton);
        backButton = findViewById(R.id.buttonBackMain);
        infoButton = findViewById(R.id.infoButtonMain);

        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            }
        });
    }
}
```

```

infoButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        View dialogView =
getLayoutInflater().inflate(R.layout.main_info_dialog, null);
        AlertDialog.Builder infoAlertDialog = new
AlertDialog.Builder(BaseActivity.this).setView(dialogView);
        infoAlertDialog.setTitle("Information");
        infoAlertDialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
int i) {
                }
            });
        infoAlertDialog.create();
        infoAlertDialog.show();
    }
});

startMeasureButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (user != null) {
            startMeasurement();
        } else {
            showNoUserInformation("Du har ingen användare, vill du
skapa en?");
        }
    }
});

showHistoryButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (user != null) {
            Intent historyIntent = new Intent(BaseActivity.this,
HistoryActivity.class);
            startActivity(historyIntent);
            finish();
        } else {
            showNoUserInformation("Du har ingen användare, vill du
skapa en?");
        }
    }
});

userButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (user != null) {
            Intent viewUserIntent = new
Intent(getApplicationContext(), ViewUserActivity.class);
            startActivity(viewUserIntent);
            finish();
        } else {
            showNoUserInformation("Du har ingen användare, vill du
skapa en?");
        }
    }
});

```

```

        }
    }
}

private void showNoUserInformation(String message) {
    AlertDialog.Builder builderNoUserDialog = new
    AlertDialog.Builder(BaseActivity.this);
    builderNoUserDialog.setTitle("Ingen användare");
    builderNoUserDialog.setMessage(message);

    builderNoUserDialog.setPositiveButton("JA", new
    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            Intent createUserIntent = new Intent(BaseActivity.this,
EditUserActivity.class);
            createUserIntent.putExtra("FROM_ACTIVITY", "MAIN");
            startActivity(createUserIntent);
            finish();
        }
    });
    builderNoUserDialog.setNegativeButton("NEJ", new
    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // User cancelled the dialog
        }
    });
}

builderNoUserDialog.create();
builderNoUserDialog.show();
}

private void startMeasurement(){
    View measureInfoDialogView =
getLayoutInflater().inflate(R.layout.measure_info_dialog, null);
    ImageView infoImage =
measureInfoDialogView.findViewById(R.id.jointInfoImage);
    TextView mainText =
measureInfoDialogView.findViewById(R.id.mainText);

    if
((user.getFingersAffected().contains("LILLFINGER") || user.getFingersAffected()
().contains("RINGFINGER")) &&
(user.getFingersAffected().contains("LÅNGFINGER") || user.getFingersAffected()
().contains("PEKFINGER"))){
        mainText.setText("Du kommer nu få ta två foton per skadat
finger\nTänk på att hålla kameran rakt framför \"lillfinger-sidan
respektive pekfingersidan\" så det blir 90° mellan kameran och handen för
korrekt mätresultat.\n\nOm en mätpunkt gömmer sig bakom ett annat finger
får du uppskatta var den punkt borde sitta.");
    } else if
((user.getFingersAffected().contains("LILLFINGER") || user.getFingersAffected()
().contains("RINGFINGER")) &&
!(user.getFingersAffected().contains("LÅNGFINGER") || user.getFingersAffected()
().contains("PEKFINGER"))){
        mainText.setText("Du kommer nu få ta två foton per skadat
finger\nTänk på att hålla kameran rakt framför \"lillfinger-sidan\" så det
blir 90° mellan kameran och handen för korrekt mätresultat.\n\nOm en
mätpunkt gömmer sig bakom ett annat finger får du uppskatta var den
punkt borde sitta.");
    } else if
}

```

```

(! (user.getFingersAffected().contains("LILLFINGER") || user.getFingersAffected()
().contains("RINGFINGER")) &&
(user.getFingersAffected().contains("LÅNGFINGER") || user.getFingersAffected()
().contains("PEKFINGER"))){
    mainText.setText("Du kommer nu få ta två foton per skadat
finger\n\nTänk på att hålla kameran rakt framför \"pekfingersidan\" så det
blir 90° mellan kameran och handen för korrekt mätresultat.\n\nOm en
mätpunkt gömmer sig bakom ett annat finger får du uppskatta vart denna
punkt borde sitta.");
}

Bitmap originalBitmap =
BitmapFactory.decodeResource(getResources(), R.drawable.joint_info);
int originalWidth = originalBitmap.getWidth();
int originalHeight = originalBitmap.getHeight();
int scaledWidth = originalWidth / 5;
int scaledHeight = originalHeight / 5;
Bitmap resizedBitmap = Bitmap.createScaledBitmap(originalBitmap,
scaledWidth, scaledHeight, false);
infoImage.setImageBitmap(resizedBitmap);

AlertDialog.Builder infoAlertBuilder = new
AlertDialog.Builder(BaseActivity.this).setView(measureInfoDialogView).setTitle(
"GÖR NY MÄTNING");
infoAlertBuilder.setPositiveButton("NÄSTA", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Intent startMeasureIntent = new Intent(BaseActivity.this,
MeasureActivity.class);
        startActivity(startMeasureIntent);
        finish();
    }
});

infoAlertBuilder.setNegativeButton("AVBRYT", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
    }
});

infoAlertBuilder.setNeutralButton("HOPPA ÖVER INFO", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Intent startViewActivityIntent = new
Intent(getApplicationContext(), MeasureActivity.class);
        startViewActivityIntent.putExtra("WantInfo", false);
        startActivity(startViewActivityIntent);
        finish();
    }
});

infoAlertBuilder.create();
infoAlertBuilder.show();

}

@Override
public void onBackPressed() {

```

```
        Intent onBackIntent = new Intent(BaseActivity.this,
BaseActivity.class); // Possibility to go back to application the
measurement tool is implemented in
        startActivity(onBackIntent);
        finish();
    }

@Override
protected void onPause() {
    saveSharedPreferences();
    super.onPause();
}

@Override
protected void onResume() {
    super.onResume();
    checkSharedPreferences();
}

protected void saveSharedPreferences() {
    String jsonString = gson.toJson(user);
    editor.putString("com.example.digitalgoniometer.user",
jsonString).commit();
}

protected void checkSharedPreferences() {
    String receivedString =
sharedPreferences.getString("com.example.digitalgoniometer.user", null);
    if (receivedString != null){
        user = gson.fromJson(receivedString, User.class);
    }
}
```

EditHistoryActivity:

```
package com.example.digitalgoniometer;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
import com.example.digitalgoniometer.model.HistoryInfo;
import java.util.ArrayList;

public class EditHistoryActivity extends HistoryActivity {

    Button historySaveButton, historyCancelButton;
    CheckBox[] checkedGraphs;
    CheckBox[] checkedFingers;

    /*
    Order of checkboxes
    [0]TAM,
    [1]MCP straight,
    [2]MCP bent,
    [3]PIP straight,
    [4]PIP bent,
    [5]DIP straight,
    [6]DIP bent,
    */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_history);

        checkedGraphs = new CheckBox[7];
        checkedFingers = new CheckBox[4];

        checkedGraphs[0] = findViewById(R.id.checkBoxHistTAM);
        checkedGraphs[1] = findViewById(R.id.checkBoxHistMCPstraight);
        checkedGraphs[2] = findViewById(R.id.checkBoxHistMCPbent);
        checkedGraphs[3] = findViewById(R.id.checkBoxHistPIPstraight);
        checkedGraphs[4] = findViewById(R.id.checkBoxHistPIPbent);
        checkedGraphs[5] = findViewById(R.id.checkBoxHistDIPstraight);
        checkedGraphs[6] = findViewById(R.id.checkBoxHistDIPbent);

        checkedFingers[0] = findViewById(R.id.checkBoxHistIndex);
        checkedFingers[1] = findViewById(R.id.checkBoxHistMiddle);
        checkedFingers[2] = findViewById(R.id.checkBoxHistRing);
        checkedFingers[3] = findViewById(R.id.checkBoxHistLittle);

        historySaveButton = findViewById(R.id.editHistSaveButton);
        historyCancelButton = findViewById(R.id.editHistCancelButton);

        checkSharedPreferencesHistory();

        if (historyInfo!= null) {

            for (int i = 0; i<historyInfo.getFingersChecked().size();i++) {
                if
(historyInfo.getFingersChecked().get(i).equals("PEKFINGER")) {
```

```

        checkedFingers[0].setChecked(true);
    }
    if
(historyInfo.getFingersChecked().get(i).equals("LÅNGFINGER")){
    checkedFingers[1].setChecked(true);
}
    if
(historyInfo.getFingersChecked().get(i).equals("RINGFINGER")){
    checkedFingers[2].setChecked(true);
}
    if
(historyInfo.getFingersChecked().get(i).equals("LILLFINGER")){
    checkedFingers[3].setChecked(true);
}
}
for (int i = 0; i<historyInfo.getGraphsChecked().size();i++) {
    if(historyInfo.getGraphsChecked().get(i).equals("TAM")){
        checkedGraphs[0].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("MCPs")){
        checkedGraphs[1].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("MCPb")){
        checkedGraphs[2].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("PIPs")){
        checkedGraphs[3].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("PIPb")){
        checkedGraphs[4].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("DIPs")){
        checkedGraphs[5].setChecked(true);
    }
    if(historyInfo.getGraphsChecked().get(i).equals("DIPb")){
        checkedGraphs[6].setChecked(true);
    }
}
}
} else{
    historyInfo = new HistoryInfo();
}

historySaveButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String[] fingers = new String[]{"PEKFINGER", "LÅNGFINGER",
"RINGFINGER", "LILLFINGER"};
        String[] graphs = new String[]{"TAM", "MCPs", "MCPb",
"PIPs", "PIPb", "DIPs", "DIPb"};
        ArrayList<String> fingersToShow = new ArrayList<>();
        ArrayList<String> graphsToShow = new ArrayList<>();

        for (int i = 0; i < checkedFingers.length; i++) {
            if (checkedFingers[i].isChecked()) {
                fingersToShow.add(fingers[i]);
            }
        }

        for(int i = 0; i<checkedGraphs.length; i++){
            if(checkedGraphs[i].isChecked()){
                graphsToShow.add(graphs[i]);
            }
        }
    }
}

```

```

        }
    }
    historyInfo.setFingersChecked(fingersToShow);
    historyInfo.setGraphsChecked(graphsToShow);
    goBack();
}
);

historyCancelButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        goBack();
    }
});

@Override
public void onBackPressed() {
    showToast();
    goBack();
}

private void showToast(){
    boolean isChecked = false;
    for(int i = 0; i< checkedGraphs.length; i++) {
        if (checkedGraphs[i].isChecked()) {
            isChecked = true;
        }
    }
    if (!isChecked) {
        Toast toast = Toast.makeText(EditHistoryActivity.this, "Inga
grafer valda!\nDu kan ändra detta under inställningar uppe i högra hörnet",
Toast.LENGTH_LONG);
        toast.show();
    }
}

private void goBack(){
    showToast();
    Intent intent = new Intent(EditHistoryActivity.this,
HistoryActivity.class);
    startActivity(intent);
    finish();
}
}

```

EditUserActivity:

```
package com.example.digitalgoniometer;

import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Switch;
import com.example.digitalgoniometer.model.User;
import java.util.ArrayList;

public class EditUserActivity extends BaseActivity {

    private Button saveEditUserButton, cancelEditUserButton, deleteButton;
    private EditText usernameInput;
    private CheckBox indexFinger, middleFinger, ringFinger, littleFinger;
    private CheckBox[] checkedFingers;
    private ArrayList<String> fingersAffected;
    private Switch leftRightSwitch;
    private String fromActivity;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_user);

        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }

        checkSharedPreferences();

        saveEditUserButton = findViewById(R.id.saveButton);
        cancelEditUserButton = findViewById(R.id.CancelButton);
        deleteButton = findViewById(R.id.removeUserButton);
        usernameInput = findViewById(R.id.userNameInputView);
        indexFinger = findViewById(R.id.checkBox1);
        middleFinger = findViewById(R.id.checkBox2);
        ringFinger = findViewById(R.id.checkBox3);
        littleFinger = findViewById(R.id.checkBox4);
        leftRightSwitch = findViewById(R.id.leftRightSwitch);

        Intent getIntent = getIntent();
        fromActivity = getIntent.getStringExtra("FROM_ACTIVITY");

        if (user != null) {
            usernameInput.setText(user.getUserName());
            leftRightSwitch.setChecked(user.isRightHand());
            for (int i = 0; i < user.getFingersAffected().size(); i++) {
                if (user.getFingersAffected().get(i).equals("PEKFINGER")) {
                    indexFinger.setChecked(true);
                }
                if (user.getFingersAffected().get(i).equals("LÅNGFINGER")) {
                    middleFinger.setChecked(true);
                }
            }
        }
    }
}
```

```

        }
        if (user.getFingersAffected().get(i).equals("RINGFINGER")) {
            ringFinger.setChecked(true);
        }
        if (user.getFingersAffected().get(i).equals("LILLFINGER")) {
            littleFinger.setChecked(true);
        }
    }
}

cancelEditUserButton.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        if (user != null){
            goBack();
        } else {
            Intent intent = new Intent(EditUserActivity.this,
BaseActivity.class);
            startActivity(intent);
            finish();
        }
    }
});
saveEditUserButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (noCheckboxChecked()) {
            giveCheckboxAlert();
        } else {

            if (user==null){
                user=new User();
            }

            if (usernameInput.getText().toString().equals("Skriv
här...")){
                user.setUserName("Användare");
            } else {

user.setUserName(usernameInput.getText().toString());
            }
            user.setRightHand(leftRightSwitch.isChecked());

            checkedFingers = new CheckBox[]{indexFinger,
middleFinger, ringFinger, littleFinger};
            String[] fingers = new String[]{"PEKFINGER",
"LÄNGFINGER", "RINGFINGER", "LILLFINGER"};
            fingersAffected = new ArrayList<>();

            for (int i = 0; i < checkedFingers.length; i++) {
                if (checkedFingers[i].isChecked()) {
                    fingersAffected.add(fingers[i]);
                }
            }
            user.setFingersAffected(fingersAffected);
            sortFingersAffectedMeasureOrder();

            goBack();
        }
    }
});

```

```

        }
    });

    deleteButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            AlertDialog.Builder builderDeleteUserDialog = new
AlertDialog.Builder(EditUserActivity.this);
            builderDeleteUserDialog.setTitle("Radera användare");
            builderDeleteUserDialog.setMessage("Är du säker på att du
vill radera din användare? All mätdata kommer försvinna.");
            builderDeleteUserDialog.setPositiveButton("JA", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    user = null;
                    Intent toMainIntent = new
Intent(EditUserActivity.this, BaseActivity.class);
                    startActivity(toMainIntent);
                    finish();
                }
            });
            builderDeleteUserDialog.setNegativeButton("NEJ", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    // User cancelled the dialog
                }
            });
            builderDeleteUserDialog.create();
            builderDeleteUserDialog.show();
        }
    });

    usernameInput.setOnFocusChangeListener(new
View.OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (hasFocus) {
                usernameInput.setText("");
            }
        }
    });
}

private void goBack(){
    Intent intent;
    if (fromActivity.equals("MAIN")){
        intent = new Intent(EditUserActivity.this, BaseActivity.class);
    } else if (fromActivity.equals("VIEW_USER")) {
        intent = new Intent(EditUserActivity.this,
ViewUserActivity.class);
    } else {
        intent = new Intent(EditUserActivity.this, BaseActivity.class);
    }
    startActivity(intent);
    finish();
}

private boolean noCheckboxChecked() {

```

```

        checkedFingers = new CheckBox[]{indexFinger, middleFinger,
ringFinger, littleFinger};
        for (int i = 0; i < checkedFingers.length; i++) {
            if (checkedFingers[i].isChecked()) {
                return false;
            }
        }
        return true;
    }

    private void giveCheckboxAlert() {
        AlertDialog.Builder builderNoCheckedDialog = new
AlertDialog.Builder(EditUserActivity.this);
        builderNoCheckedDialog.setTitle("Inget finger valt");
        builderNoCheckedDialog.setMessage("Du måste välja minst ett skadat
finger");

        builderNoCheckedDialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
            }
        });

        builderNoCheckedDialog.create();
        builderNoCheckedDialog.show();
    }

    private void sortFingersAffectedMeasureOrder() {
        ArrayList<String> tempArr = new ArrayList<>();
        if (user.getFingersAffected().contains("LILLFINGER")){
            tempArr.add("LILLFINGER");
        }
        if (user.getFingersAffected().contains("RINGFINGER")){
            tempArr.add("RINGFINGER");
        }
        if (user.getFingersAffected().contains("PEKFINGER")){
            tempArr.add("PEKFINGER");
        }
        if (user.getFingersAffected().contains("LÅNGFINGER")){
            tempArr.add("LÅNGFINGER");
        }
        user.setFingersAffected(tempArr);
    }

    @Override
    public void onBackPressed() {
        goBack();
    }
}

```

HistoryActivity:

```
package com.example.digitalgoniometer;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget ScrollView;
import android.widget.Switch;
import android.widget.TableLayout;
import android.graphics.DashPathEffect;
import android.widget.TableRow;
import android.widget.TextView;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import com.androidplot.xy.BoundaryMode;
import com.androidplot.xy.LineAndPointFormatter;
import com.androidplot.xy.SimpleXYSeries;
import com.androidplot.xy.StepMode;
import com.androidplot.xy.XYGraphWidget;
import com.androidplot.xy.XYPlot;
import com.androidplot.xy.XYSeries;
import com.example.digitalgoniometer.model.DateFormatter;
import com.example.digitalgoniometer.model.HistoryInfo;
import com.example.digitalgoniometer.model.User;
import java.text.DecimalFormat;
import java.util.Arrays;

public class HistoryActivity extends BaseActivity {

    private XYPlot plot;
    private TableLayout table;
    private Button backButton, editHistoryButton;
    private String[] xLabels;
    protected HistoryInfo historyInfo;
    private LinearLayout graphInfo1;
    private androidx.appcompat.widget.LinearLayoutCompat graphInfo2;
    private ScrollView scrollView;
    private boolean historyinfoExisted;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        checkSharedPreferencesHistory();

        if (historyInfo == null) {
            historyInfo = new HistoryInfo();
            historyinfoExisted = false;
        }
    }
}
```

```

    } else {
        historyinfoExisted = true;
    }

    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        actionBar.hide();
    }

    backButton = findViewById(R.id.historyBackButton);
    editHistoryButton = findViewById(R.id.editHistoryButton);
    plot = findViewById(R.id.graphViewPlot);
    table = findViewById(R.id.tableLayout);
    graphInfo1 = findViewById(R.id.colorLinearLayout);
    graphInfo2 = findViewById(R.id.colorLinearLayout2);
    scrollView = findViewById(R.id.scrollViewTable);
    Switch switchView = findViewById(R.id.leftRightSwitchHistory);

    switchView.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            if (isChecked) {
                table.setVisibility(View.GONE);
                plot.setVisibility(View.VISIBLE);
                graphInfo1.setVisibility(View.VISIBLE);
                graphInfo2.setVisibility(View.VISIBLE);
            } else {
                table.setVisibility(View.VISIBLE);
                scrollView.setVisibility(View.VISIBLE);
                plot.setVisibility(View.GONE);
                graphInfo1.setVisibility(View.GONE);
                graphInfo2.setVisibility(View.GONE);
            }
        }
    });
}

updateTable();

plot.setRangeBoundaries(-10, 200, BoundaryMode.FIXED);
plot.getGraph().getBackgroundPaint().setColor(Color.TRANSPARENT);
plot.getBackgroundPaint().setColor(Color.TRANSPARENT);
plot.redraw();
getValuesForPlot(user.getFingersAffected().get(0));

backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent backIntent = new Intent(HistoryActivity.this,
 BaseActivity.class);
        startActivity(backIntent);
        finish();
    }
});

editHistoryButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent editIntent = new Intent(HistoryActivity.this,
 EditHistoryActivity.class);

```

```

        startActivity(editIntent);
        finish();
    });
}
if (!historyinfoExisted) {
    showNoGraphSelectedInformation();
}

private void updateTable(){
    String[] headings = { " Datum:\n      ", " TAM\n      ", " DIP\nböjd", " DIP\n sträckt", " PIP\n böjd", " PIP\n sträckt", " MCP\n böjd", " MCP\n sträckt" };
    TableRow headerRow = new TableRow(this);

    for (int i = 0; i < headings.length; i++) {
        TextView textView = new TextView(this);
        textView.setText(headings[i]);

        WindowManager windowManager = (WindowManager)
this.getSystemService(Context.WINDOW_SERVICE);
        DisplayMetrics displayMetrics = new DisplayMetrics();
        windowManager.getDefaultDisplay().getMetrics(displayMetrics);
        int screenWidth = displayMetrics.widthPixels;

        if (i==0){
            textView.setWidth(screenWidth/8);
        } else {
            textView.setWidth(screenWidth/8);
        }

        textView.setBackgroundColor(getResources().getColor(R.color.stockholm_blue));
        textView.setTextColor(getResources().getColor(R.color.white));
        textView.setTextSize(16);

        headerRow.addView(textView);
    }
    table.addView(headerRow);

    for (int i = 0; i< user.getMeasurementsLittle().size(); i++){
        TableRow row = new TableRow(this);
        TextView dateTextView = new TextView(this);

        dateTextView.setText(user.getMeasurementsLittle().get(i).getDate());
        TextView tamTextView = new TextView(this);

        tamTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).ge
tTamValue()));
        TextView dipbTextView = new TextView(this);

        dipbTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).g
etDipAngleBent()));
        TextView dipsTextView = new TextView(this);

        dipsTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).g
etDipAngleStraight()));
        TextView pipbTextView = new TextView(this);

        pipbTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).g
etDipAngleStraight()));
    }
}

```

```

        pipbTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).getPipAngleBent()));
        TextView pipsTextView = new TextView(this);

        pipsTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).getDipAngleStraight()));
        TextView mcpbTextView = new TextView(this);

        mcpbTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).getMcpAngleBent()));
        TextView mcpsTextView = new TextView(this);

        mcpsTextView.setText(Integer.toString(user.getMeasurementsLittle().get(i).getDipAngleStraight()));

        if((i%2) == 0) {

            dateTextView.setBackgroundColor(getResources().getColor(R.color.white));
            tamTextView.setBackgroundColor(getResources().getColor(R.color.white));
            dipbTextView.setBackgroundColor(getResources().getColor(R.color.white));
            dipsTextView.setBackgroundColor(getResources().getColor(R.color.white));
            pipbTextView.setBackgroundColor(getResources().getColor(R.color.white));
            pipsTextView.setBackgroundColor(getResources().getColor(R.color.white));
            mcpbTextView.setBackgroundColor(getResources().getColor(R.color.white));
            mcpsTextView.setBackgroundColor(getResources().getColor(R.color.white));
        }

            dateTextView.setTextSize(14);
            tamTextView.setTextSize(14);
            dipbTextView.setTextSize(14);
            dipsTextView.setTextSize(14);
            pipbTextView.setTextSize(14);
            pipsTextView.setTextSize(14);
            mcpbTextView.setTextSize(14);
            mcpsTextView.setTextSize(14);
            row.addView(dateTextView);
            row.addView(tamTextView);
            row.addView(dipbTextView);
            row.addView(dipsTextView);
            row.addView(pipbTextView);
            row.addView(pipsTextView);
            row.addView(mcpbTextView);
            row.addView(mcpsTextView);

            table.addView(row);
        }
    }

private void formatLegend() {
    plot.getLayoutManager().remove(plot.getLegend());

    ImageView tamColorView = findViewById(R.id.tamColorView);
    ImageView dipColorView = findViewById(R.id.dipColorView);
    ImageView pipColorView = findViewById(R.id.pipColorView);
}

```

```

        ImageView mcpColorView = findViewById(R.id.mcpColorView);
        ImageView bentLineView = findViewById(R.id.bentLineView);
        ImageView straightLineView = findViewById(R.id.straightLineView);

        Drawable circleDrawableTam =
getResources().getDrawable(android.R.drawable.presence_online).getConstantS
tate().newDrawable().mutate();
        Drawable circleDrawableDip =
getResources().getDrawable(android.R.drawable.presence_online).getConstantS
tate().newDrawable().mutate();
        Drawable circleDrawablePip =
getResources().getDrawable(android.R.drawable.presence_online).getConstantS
tate().newDrawable().mutate();
        Drawable circleDrawableMcp =
getResources().getDrawable(android.R.drawable.presence_online).getConstantS
tate().newDrawable().mutate();
        Drawable drawableLine =
getResources().getDrawable(R.drawable.line);
        Drawable drawableDashedLine =
getResources().getDrawable(R.drawable.dashed_line);

        circleDrawableTam.setColorFilter(Color.BLACK,
PorterDuff.Mode.SRC_IN);
        circleDrawableMcp.setColorFilter(Color.BLUE,
PorterDuff.Mode.SRC_IN);
        circleDrawableDip.setColorFilter(Color.parseColor("#FFC900"),
PorterDuff.Mode.SRC_IN);
        circleDrawablePip.setColorFilter(Color.parseColor("#FFBB86FC"),
PorterDuff.Mode.SRC_IN);

        tamColorView.setImageDrawable(circleDrawableTam);
        dipColorView.setImageDrawable(circleDrawableDip);
        pipColorView.setImageDrawable(circleDrawablePip);
        mcpColorView.setImageDrawable(circleDrawableMcp);
        bentLineView.setImageDrawable(drawableLine);
        straightLineView.setImageDrawable(drawableDashedLine);

    }

private void createLabels(String finger) {

    switch (finger) {

        case "LILLFINGER":
            xLabels = new String[user.getMeasurementsLittle().size() +
1];
            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    xLabels[0] = "Start";
                    xLabels[1] =
user.getMeasurementsLittle().get(0).getDate().substring(5);
                } else {
                    xLabels[i] = user.getMeasurementsLittle().get(i -
1).getDate().substring(5);
                }
            }
            break;

        case "RINGFINGER":
            xLabels = new String[user.getMeasurementsRing().size() +

```

```

1];
        for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
            if (i == 1) {
                xLabels[0] = "Start";
                xLabels[1] =
user.getMeasurementsRing().get(0).getDate().substring(5);
            } else {
                xLabels[i] = user.getMeasurementsRing().get(i -
1).getDate().substring(5);
            }
        }
        break;

    case "LÅNGFINGER":
        xLabels = new String[user.getMeasurementsMiddle().size() +
1];
        for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
            if (i == 1) {
                xLabels[0] = "Start";
                xLabels[1] =
user.getMeasurementsMiddle().get(0).getDate().substring(5);
            } else {
                xLabels[i] = user.getMeasurementsMiddle().get(i -
1).getDate().substring(5);
            }
        }
        break;

    case "PEKFINGER":
        xLabels = new String[user.getMeasurementsIndex().size() +
1];
        for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
            if (i == 1) {
                xLabels[0] = "Start";
                xLabels[1] =
user.getMeasurementsIndex().get(0).getDate().substring(5);
            } else {
                xLabels[i] = user.getMeasurementsIndex().get(i -
1).getDate().substring(5);
            }
        }
        break;
    }

}

private void formatAxis() {

    plot.setDomainStepMode(StepMode.INCREMENT_BY_VAL);

plot.getGraph().getLineLabelStyle(XYGraphWidget.Edge.BOTTOM).setFormat(new
DateFormatter(xLabels));

    plot.setDomainStepValue(5);
    if (xLabels.length<=70){
        plot.setDomainStepValue(4);
    }
    if(xLabels.length<=50){


```

```

        plot.setDomainStepValue(3);
    }
    if (xLabels.length<=30) {
        plot.setDomainStepValue(2);
    }
    if (xLabels.length<=10) {
        plot.setDomainStepValue(1);
    }

    plot.setRangeStep(StepMode.INCREMENT_BY_VAL, 10);
    DecimalFormat format = new DecimalFormat("#");

plot.getGraph().getLineLabelStyle(XYGraphWidget.Edge.LEFT).setFormat(format);
}

private void addSeriesToPlot() {
    if (!historyInfo.getFingersChecked().isEmpty() ||
!historyInfo.getGraphsChecked().isEmpty()) {
        for (int i = 0; i < historyInfo.getFingersChecked().size();
i++) {
            switch (historyInfo.getFingersChecked().get(i)) {

                case "PEKFINGER":
                    for (int j = 0; j <
historyInfo.getGraphsChecked().size(); j++) {
                        switch (historyInfo.getGraphsChecked().get(j))
{

                            case "TAM":
                                XYSeries seriesTAMIndex = new
SimpleXYSeries(NSArray.asList(getTamDataPoints("PEKFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: TAM ");
                                LineAndPointFormatter formatTamIndex =
new LineAndPointFormatter(Color.BLACK, Color.BLACK, null, null);
                                plot.addSeries(seriesTAMIndex,
formatTamIndex);
                                break;

                            case "MCPs":
                                XYSeries seriesMCPsIndex = new
SimpleXYSeries(NSArray.asList(getMCPsDataPoints("PEKFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: MCP-rak");
                                LineAndPointFormatter formatMCPsIndex =
new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);
                                formatMCPsIndex.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
                                plot.addSeries(seriesMCPsIndex,
formatMCPsIndex);
                                break;
                            case "MCPb":
                                XYSeries seriesMCPbIndex = new
SimpleXYSeries(NSArray.asList(getMCPbDataPoints("PEKFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: MCP-böjd");
                                LineAndPointFormatter formatMCPbIndex =
new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);
                                plot.addSeries(seriesMCPbIndex,
formatMCPbIndex);
                                break;
                            case "PIPs":
```

```

        XYSeries seriesPIPsIndex = new
SimpleXYSeries(NSArray.asList(getPIPsDataPoints("PEKFINGER"))),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: PIP-rak");
LineAndPointFormatter formatPIPsIndex =
new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK, null,
null);

formatPIPsIndex.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesPIPsIndex,
formatPIPsIndex);
break;
case "PIPb":
XYSeries seriesPIPbIndex = new
SimpleXYSeries(NSArray.asList(getPIPbDataPoints("PEKFINGER"))),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: PIP-böjd");
LineAndPointFormatter formatPIPbIndex =
new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK, null,
null);
plot.addSeries(seriesPIPbIndex,
formatPIPbIndex);
break;

case "DIPs":
XYSeries seriesDIPsIndex = new
SimpleXYSeries(NSArray.asList(getDIPsDataPoints("PEKFINGER"))),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: DIP-rak");
LineAndPointFormatter formatDIPsIndex =
new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);

formatDIPsIndex.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesDIPsIndex,
formatDIPsIndex);
break;
case "DIPb":
XYSeries seriesDIPbIndex = new
SimpleXYSeries(NSArray.asList(getDIPbDataPoints("PEKFINGER"))),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Pek: DIP-böjd");
LineAndPointFormatter formatDIPbIndex =
new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);
plot.addSeries(seriesDIPbIndex,
formatDIPbIndex);
break;
}
}
break;

case "LÅNGFINGER":
for (int j = 0; j <
historyInfo.getGraphsChecked().size(); j++) {
switch (historyInfo.getGraphsChecked().get(j))
{
case "TAM":
XYSeries seriesTAMmiddle = new
SimpleXYSeries(NSArray.asList(getTamDataPoints("LÅNGFINGER"))),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: TAM");

```

```

        LineAndPointFormatter formatTamMiddle =
new LineAndPointFormatter(Color.BLACK, Color.BLACK, null, null);
plot.addSeries(seriesTAMmiddle,
formatTamMiddle);
break;

case "MCPs":
XYSeries seriesMCPsMiddle = new
SimpleXYSeries(Arrays.asList(getMCPsDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: MCP-rak");
LineAndPointFormatter formatMCPsMiddle
= new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);

formatMCPsMiddle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesMCPsMiddle,
formatMCPsMiddle);
break;
case "MCPb":
XYSeries seriesMCPbMiddle = new
SimpleXYSeries(Arrays.asList(getMCPbDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: MCP-böjd");
LineAndPointFormatter formatMCPbMiddle
= new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);
plot.addSeries(seriesMCPbMiddle,
formatMCPbMiddle);
break;
case "PIPs":
XYSeries seriesPIPsMiddle = new
SimpleXYSeries(Arrays.asList(getPIPsDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: PIP-rak");
LineAndPointFormatter formatPIPsMiddle
= new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK,
null, null);

formatPIPsMiddle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesPIPsMiddle,
formatPIPsMiddle);
break;
case "PIPb":
XYSeries seriesPIPbMiddle = new
SimpleXYSeries(Arrays.asList(getPIPbDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: PIP-böjd");
LineAndPointFormatter formatPIPbMiddle
= new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK,
null, null);
plot.addSeries(seriesPIPbMiddle,
formatPIPbMiddle);
break;
case "DIPs":
XYSeries seriesDIPsMiddle = new
SimpleXYSeries(Arrays.asList(getDIPsDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: DIP-rak");
LineAndPointFormatter formatDIPsMiddle
= new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);

formatDIPsMiddle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesDIPsMiddle,

```

```

formatDIPsMiddle);
                                break;
                case "DIPb":
                        XYSeries seriesDIPbMiddle = new
SimpleXYSeries(NSArray.asList(getDIPbDataPoints("LÄNGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lång: DIP-böjd");
                        LineAndPointFormatter formatDIPbMiddle
= new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);
                        plot.addSeries(seriesDIPbMiddle,
formatDIPbMiddle);
                                break;
}
        }
        break;

        case "RINGFINGER":
                for (int j = 0; j <
historyInfo.getGraphsChecked().size(); j++) {
                        switch (historyInfo.getGraphsChecked().get(j))
{
                        }

                        case "TAM":
                                XYSeries seriesTAMring = new
SimpleXYSeries(NSArray.asList(getTamDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: TAM");
                                LineAndPointFormatter formatTamRing =
new LineAndPointFormatter(Color.BLACK, Color.BLACK, null, null);
                                plot.addSeries(seriesTAMring,
formatTamRing);
                                break;

                        case "MCPs":
                                XYSeries seriesMCPsRing = new
SimpleXYSeries(NSArray.asList(getMCPsDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: MCP-rak");
                                LineAndPointFormatter formatMCPsRing =
new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);

formatMCPsRing.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
                                plot.addSeries(seriesMCPsRing,
formatMCPsRing);
                                break;
                        case "MCPb":
                                XYSeries seriesMCPbRing = new
SimpleXYSeries(NSArray.asList(getMCPbDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: MCP-böjd");
                                LineAndPointFormatter formatMCPbRing =
new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);
                                plot.addSeries(seriesMCPbRing,
formatMCPbRing);
                                break;
                        case "PIPs":
                                XYSeries seriesPIPsRing = new
SimpleXYSeries(NSArray.asList(getPIPsDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: PIP-rak");
                                LineAndPointFormatter formatPIPsRing =
new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK, null,
null);

```

```

formatPIPsRing.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesPIPsRing,
formatPIPsRing);
break;
case "PIPb":
XYSeries seriesPIPbRing = new
SimpleXYSeries(NSArray.asList(getPIPbDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: PIP-böjd");
LineAndPointFormatter formatPIPbRing =
new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK, null,
null);
plot.addSeries(seriesPIPbRing,
formatPIPbRing);
break;
case "DIPs":
XYSeries seriesDIPsRing = new
SimpleXYSeries(NSArray.asList(getDIPsDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: DIP-rak");
LineAndPointFormatter formatDIPsRing =
new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);

formatDIPsRing.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesDIPsRing,
formatDIPsRing);
break;
case "DIPb":
XYSeries seriesDIPbRing = new
SimpleXYSeries(NSArray.asList(getDIPbDataPoints("RINGFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Ring: DIP-böjd");
LineAndPointFormatter formatDIPbRing =
new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);
plot.addSeries(seriesDIPbRing,
formatDIPbRing);
break;
}
}
break;

case "LILLFINGER":
for (int j = 0; j <
historyInfo.getGraphsChecked().size(); j++) {
switch (historyInfo.getGraphsChecked().get(j))
{
case "TAM":
XYSeries seriesTAMlittle = new
SimpleXYSeries(NSArray.asList(getTAMDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: TAM");
LineAndPointFormatter formatTamLittle =
new LineAndPointFormatter(Color.BLACK, Color.BLACK, null, null);
plot.addSeries(seriesTAMlittle,
formatTamLittle);
break;

case "MCPs":
XYSeries seriesMCPsLittle = new
SimpleXYSeries(NSArray.asList(getMCPsDataPoints("LILLFINGER")),

```

```

SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: MCP-rak");
LineAndPointFormatter formatMCPsLittle
= new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);

formatMCPsLittle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesMCPsLittle,
formatMCPsLittle);
break;
case "MCPb":
XYSeries seriesMCPbLittle = new
SimpleXYSeries(NSArray.asList(getMCPbDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: MCP-böjd");
LineAndPointFormatter formatMCPbLittle
= new LineAndPointFormatter(Color.BLUE, Color.BLACK, null, null);
plot.addSeries(seriesMCPbLittle,
formatMCPbLittle);
break;
case "PIPs":
XYSeries seriesPIPsLittle = new
SimpleXYSeries(NSArray.asList(getPIPsDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: PIP-rak");
LineAndPointFormatter formatPIPsLittle
= new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK,
null, null);

formatPIPsLittle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesPIPsLittle,
formatPIPsLittle);
break;
case "PIPb":
XYSeries seriesPIPbLittle = new
SimpleXYSeries(NSArray.asList(getPIPbDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: PIP-böjd");
LineAndPointFormatter formatPIPbLittle
= new LineAndPointFormatter(Color.parseColor("#FFBB86FC"), Color.BLACK,
null, null);
plot.addSeries(seriesPIPbLittle,
formatPIPbLittle);
break;
case "DIPs":
XYSeries seriesDIPsLittle = new
SimpleXYSeries(NSArray.asList(getDIPsDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: DIP-rak");
LineAndPointFormatter formatDIPsLittle
= new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);

formatDIPsLittle.getLinePaint().setPathEffect(new DashPathEffect(new
float[]{10, 10}, 0));
plot.addSeries(seriesDIPsLittle,
formatDIPsLittle);
break;
case "DIPb":
XYSeries seriesDIPbLittle = new
SimpleXYSeries(NSArray.asList(getDIPbDataPoints("LILLFINGER")),
SimpleXYSeries.ArrayFormat.XY_VALS_INTERLEAVED, "Lill: DIP-böjd");
LineAndPointFormatter formatDIPbLittle
= new LineAndPointFormatter(Color.parseColor("#FFC900"), Color.BLACK, null,
null);

```

```

        plot.addSeries(seriesDIPbLittle,
formatDIPbLittle);
        break;
    }
}
break;
}
}
}

plot.redraw();
}

private void showNoGraphSelectedInformation() {
    AlertDialog.Builder builderNoGraphDialog = new
AlertDialog.Builder(HistoryActivity.this);
    builderNoGraphDialog.setTitle("Välj graf att visa");
    builderNoGraphDialog.setMessage("Du behöver välja grafer som ska
visas under Inställningar");

    builderNoGraphDialog.setPositiveButton("Inställningar", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            Intent chooseGraphIntent = new Intent(HistoryActivity.this,
EditHistoryActivity.class);
            startActivity(chooseGraphIntent);
            finish();
        }
    });
    builderNoGraphDialog.setNegativeButton("Avbryt", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
        }
    });
    builderNoGraphDialog.create();
    builderNoGraphDialog.show();
}

private void getValuesForPlot(String finger) {
    createLabels(finger);
    formatAxis();
    formatLegend();
    addSeriesToPlot();
}

private Number[] getTAMDataPoints(String finger) {
    Number[] dataPointsTAM = null;
    switch (finger) {
        case "LILLFINGER":
            dataPointsTAM = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];
            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsTAM[0] = 0;
                    dataPointsTAM[2] = 1;
                } else {
                    dataPointsTAM[i * 2] = i;
                }
            }
    }
}

```

```

        }
    }

    for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
        if (i == 1) {
            dataPointsTAM[1] = 0;
            dataPointsTAM[3] =
user.getMeasurementsLittle().get(0).getTamValue();
        } else {
            dataPointsTAM[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getTamValue();
        }
        break;
    case "RINGFINGER":
        dataPointsTAM = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];
        for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
            if (i == 1) {
                dataPointsTAM[0] = 0;
                dataPointsTAM[2] = 1;
            } else {
                dataPointsTAM[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
            if (i == 1) {
                dataPointsTAM[1] = 0;
                dataPointsTAM[3] =
user.getMeasurementsRing().get(0).getTamValue();
            } else {
                dataPointsTAM[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getTamValue();
            }
            break;
        case "LÄNGFINGER":
            dataPointsTAM = new
Number[ (user.getMeasurementsMiddle().size() * 2) + 2];
            for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsTAM[0] = 0;
                    dataPointsTAM[2] = 1;
                } else {
                    dataPointsTAM[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
                if (i == 1) {
                    dataPointsTAM[1] = 0;

```

```

        dataPointsTAM[3] =
user.getMeasurementsMiddle().get(0).getTamValue();
    } else {
        dataPointsTAM[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getTamValue();
    }
}
break;
case "PEKFINGER":

    dataPointsTAM = new
Number[ (user.getMeasurementsIndex().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
        if (i == 1) {
            dataPointsTAM[0] = 0;
            dataPointsTAM[2] = 1;
        } else {
            dataPointsTAM[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
        if (i == 1) {
            dataPointsTAM[1] = 0;
            dataPointsTAM[3] =
user.getMeasurementsIndex().get(0).getTamValue();
        } else {
            dataPointsTAM[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getTamValue();
        }
    }
    break;
}
return dataPointsTAM;
}

private Number[] getMCPsDataPoints(String finger) {
    Number[] dataPointsMCPs = null;
    switch (finger) {
        case "LILLFINGER":

            dataPointsMCPs = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsMCPs[0] = 0;
                    dataPointsMCPs[2] = 1;
                } else {
                    dataPointsMCPs[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                if (i == 1) {

```

```

        dataPointsMCps[1] = 0;
        dataPointsMCps[3] =
user.getMeasurementsLittle().get(0).getMcpAngleStraight();
    } else {
        dataPointsMCps[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getMcpAngleStraight();
    }
}
break;

case "RINGFINGER":
    dataPointsMCps = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
        if (i == 1) {
            dataPointsMCps[0] = 0;
            dataPointsMCps[2] = 1;
        } else {
            dataPointsMCps[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
        if (i == 1) {
            dataPointsMCps[1] = 0;
            dataPointsMCps[3] =
user.getMeasurementsRing().get(0).getMcpAngleStraight();
        } else {
            dataPointsMCps[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getMcpAngleStraight();
        }
    }
}
break;
case "LÅNGFINGER":
    dataPointsMCps = new
Number[ (user.getMeasurementsMiddle().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
        if (i == 1) {
            dataPointsMCps[0] = 0;
            dataPointsMCps[2] = 1;
        } else {
            dataPointsMCps[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
        if (i == 1) {
            dataPointsMCps[1] = 0;
            dataPointsMCps[3] =
user.getMeasurementsMiddle().get(0).getMcpAngleStraight();
        } else {
            dataPointsMCps[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getMcpAngleStraight();
        }
    }
}

```

```

        }
    }
    break;
case "PEKFINGER":

    dataPointsMCPs = new
Number[ (user.getMeasurementsIndex().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
        if (i == 1) {
            dataPointsMCPs[0] = 0;
            dataPointsMCPs[2] = 1;
        } else {
            dataPointsMCPs[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
        if (i == 1) {
            dataPointsMCPs[1] = 0;
            dataPointsMCPs[3] =
user.getMeasurementsIndex().get(0).getMcpAngleStraight();
        } else {
            dataPointsMCPs[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getMcpAngleStraight();
        }
    }
    break;
}
return dataPointsMCPs;
}

private Number[] getMCPbDataPoints(String finger) {
    Number[] dataPointsMCPb = null;
    switch (finger) {
        case "LILLFINGER":

            dataPointsMCPb = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsMCPb[0] = 0;
                    dataPointsMCPb[2] = 1;
                } else {
                    dataPointsMCPb[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                if (i == 1) {
                    dataPointsMCPb[1] = 0;
                    dataPointsMCPb[3] =
user.getMeasurementsLittle().get(0).getMcpAngleBent();
                } else {
                    dataPointsMCPb[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getMcpAngleBent();
                }
            }
    }
}

```

```

        }
    }
    break;

case "RINGFINGER":

    dataPointsMCPb = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
        if (i == 1) {
            dataPointsMCPb[0] = 0;
            dataPointsMCPb[2] = 1;
        } else {
            dataPointsMCPb[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
        if (i == 1) {
            dataPointsMCPb[1] = 0;
            dataPointsMCPb[3] =
user.getMeasurementsRing().get(0).getMcpAngleBent();
        } else {
            dataPointsMCPb[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getMcpAngleBent();
        }
    }
    break;
case "LÅNGFINGER":

    dataPointsMCPb = new
Number[ (user.getMeasurementsMiddle().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
        if (i == 1) {
            dataPointsMCPb[0] = 0;
            dataPointsMCPb[2] = 1;
        } else {
            dataPointsMCPb[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
        if (i == 1) {
            dataPointsMCPb[1] = 0;
            dataPointsMCPb[3] =
user.getMeasurementsMiddle().get(0).getMcpAngleBent();
        } else {
            dataPointsMCPb[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getMcpAngleBent();
        }
    }
    break;
case "PEKFINGER":

    dataPointsMCPb = new

```

```

Number[ (user.getMeasurementsIndex().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
            if (i == 1) {
                dataPointsMCPb[0] = 0;
                dataPointsMCPb[2] = 1;
            } else {
                dataPointsMCPb[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
            if (i == 1) {
                dataPointsMCPb[1] =
0;//user.measurements().get(0).getTamValue();
                dataPointsMCPb[3] =
user.getMeasurementsIndex().get(0).getMcpAngleBent();
            } else {
                dataPointsMCPb[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getMcpAngleBent();
            }
        }
        break;
    }
    return dataPointsMCPb;
}

private Number[] getPIPsDataPoints(String finger) {
    Number[] dataPointsPIPs = null;
    switch (finger) {
        case "LILLFINGER":

            dataPointsPIPs = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsPIPs[0] = 0;
                    dataPointsPIPs[2] = 1;
                } else {
                    dataPointsPIPs[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                if (i == 1) {
                    dataPointsPIPs[1] = 0;
                    dataPointsPIPs[3] =
user.getMeasurementsLittle().get(0).getPipAngleStraight();
                } else {
                    dataPointsPIPs[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getPipAngleStraight();
                }
            }
            break;

        case "RINGFINGER":
    }
}

```

```

        dataPointsPIPs = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
            if (i == 1) {
                dataPointsPIPs[0] = 0;
                dataPointsPIPs[2] = 1;
            } else {
                dataPointsPIPs[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
            if (i == 1) {
                dataPointsPIPs[1] = 0;
                dataPointsPIPs[3] =
user.getMeasurementsRing().get(0).getPipAngleStraight();
            } else {
                dataPointsPIPs[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getPipAngleStraight();
            }
        }
        break;
    case "LÅNGFINGER":

        dataPointsPIPs = new
Number[ (user.getMeasurementsMiddle().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
            if (i == 1) {
                dataPointsPIPs[0] = 0;
                dataPointsPIPs[2] = 1;
            } else {
                dataPointsPIPs[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
            if (i == 1) {
                dataPointsPIPs[1] = 0;
                dataPointsPIPs[3] =
user.getMeasurementsMiddle().get(0).getPipAngleStraight();
            } else {
                dataPointsPIPs[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getPipAngleStraight();
            }
        }
        break;
    case "PEKFINGER":

        dataPointsPIPs = new
Number[ (user.getMeasurementsIndex().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
            if (i == 1) {

```

```

        dataPointsPIPs[0] = 0;
        dataPointsPIPs[2] = 1;
    } else {
        dataPointsPIPs[i * 2] = i;
    }
}

for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
    if (i == 1) {
        dataPointsPIPs[1] = 0;
        dataPointsPIPs[3] =
user.getMeasurementsIndex().get(0).getPipAngleStraight();
    } else {
        dataPointsPIPs[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getPipAngleStraight();
    }
    break;
}
return dataPointsPIPs;
}

private Number[] getPIPbDataPoints(String finger) {

    Number[] dataPointsPIPb = null;
    switch (finger) {
        case "LILLFINGER":

            dataPointsPIPb = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsPIPb[0] = 0;
                    dataPointsPIPb[2] = 1;
                } else {
                    dataPointsPIPb[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                if (i == 1) {
                    dataPointsPIPb[1] = 0;
                    dataPointsPIPb[3] =
user.getMeasurementsLittle().get(0).getPipAngleBent();
                } else {
                    dataPointsPIPb[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getPipAngleBent();
                }
            }
            break;

        case "RINGFINGER":

            dataPointsPIPb = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];
            for (int i = 1; i < user.getMeasurementsRing().size() + 1;

```

```

i++) {
    if (i == 1) {
        dataPointsPIPb[0] = 0;
        dataPointsPIPb[2] = 1;
    } else {
        dataPointsPIPb[i * 2] = i;
    }
}

for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
    if (i == 1) {
        dataPointsPIPb[1] = 0;
        dataPointsPIPb[3] =
user.getMeasurementsRing().get(0).getPipAngleBent();
    } else {
        dataPointsPIPb[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getPipAngleBent();
    }
}
break;

case "LÅNGFINGER":

    dataPointsPIPb = new
Number[ (user.getMeasurementsMiddle().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
        if (i == 1) {
            dataPointsPIPb[0] = 0;
            dataPointsPIPb[2] = 1;
        } else {
            dataPointsPIPb[i * 2] = i;
        }
    }

    for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
        if (i == 1) {
            dataPointsPIPb[1] = 0;
            dataPointsPIPb[3] =
user.getMeasurementsMiddle().get(0).getPipAngleBent();
        } else {
            dataPointsPIPb[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getPipAngleBent();
        }
    }
}
break;

case "PEKFINGER":

    dataPointsPIPb = new
Number[ (user.getMeasurementsIndex().size() * 2) + 2];

    for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
        if (i == 1) {
            dataPointsPIPb[0] = 0;
            dataPointsPIPb[2] = 1;
        } else {

```

```

                dataPointsPIPb[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
            if (i == 1) {
                dataPointsPIPb[1] = 0;
                dataPointsPIPb[3] =
user.getMeasurementsIndex().get(0).getPipAngleBent();
            } else {
                dataPointsPIPb[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getPipAngleBent();
            }
            break;
        }
        return dataPointsPIPb;
    }

    private Number[] getDIPsDataPoints(String finger) {

        Number[] dataPointsDIPs = null;
        switch (finger) {
            case "LILLFINGER":

                dataPointsDIPs = new
Number[ (user.getMeasurementsLittle().size() * 2) + 2];

                for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                    if (i == 1) {
                        dataPointsDIPs[0] = 0;
                        dataPointsDIPs[2] = 1;
                    } else {
                        dataPointsDIPs[i * 2] = i;
                    }
                }

                for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                    if (i == 1) {
                        dataPointsDIPs[1] = 0;
                        dataPointsDIPs[3] =
user.getMeasurementsLittle().get(0).getDipAngleStraight();
                    } else {
                        dataPointsDIPs[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getDipAngleStraight();
                    }
                }
                break;

            case "RINGFINGER":

                dataPointsDIPs = new
Number[ (user.getMeasurementsRing().size() * 2) + 2];

                for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
                    if (i == 1) {
                        dataPointsDIPs[0] = 0;

```

```

                dataPointsDIPs[2] = 1;
            } else {
                dataPointsDIPs[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
            if (i == 1) {
                dataPointsDIPs[1] = 0;
                dataPointsDIPs[3] =
user.getMeasurementsRing().get(0).getDipAngleStraight();
            } else {
                dataPointsDIPs[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getDipAngleStraight();
            }
        }
        break;
    case "LÅNGFINGER":

        dataPointsDIPs = new
Number[(user.getMeasurementsMiddle().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
            if (i == 1) {
                dataPointsDIPs[0] = 0;
                dataPointsDIPs[2] = 1;
            } else {
                dataPointsDIPs[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
            if (i == 1) {
                dataPointsDIPs[1] = 0;
                dataPointsDIPs[3] =
user.getMeasurementsMiddle().get(0).getDipAngleStraight();
            } else {
                dataPointsDIPs[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getDipAngleStraight();
            }
        }
        break;

    case "PEKFINGER":

        dataPointsDIPs = new
Number[(user.getMeasurementsIndex().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
            if (i == 1) {
                dataPointsDIPs[0] = 0;
                dataPointsDIPs[2] = 1;
            } else {
                dataPointsDIPs[i * 2] = i;
            }
        }
    }
}

```

```

        for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
            if (i == 1) {
                dataPointsDIPs[1] = 0;
                dataPointsDIPs[3] =
user.getMeasurementsIndex().get(0).getDipAngleStraight();
            } else {
                dataPointsDIPs[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getDipAngleStraight();
            }
        }
        break;
    }
    return dataPointsDIPs;
}

private Number[] getDIPbDataPoints(String finger) {

    Number[] dataPointsDIPb = null;
    switch (finger) {
        case "LILLFINGER":

            dataPointsDIPb = new
Number[(user.getMeasurementsLittle().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsLittle().size() +
1; i++) {
                if (i == 1) {
                    dataPointsDIPb[0] = 0;
                    dataPointsDIPb[2] = 1;
                } else {
                    dataPointsDIPb[i * 2] = i;
                }
            }

            for (int i = 1; i < (user.getMeasurementsLittle().size() +
1); i++) {
                if (i == 1) {
                    dataPointsDIPb[1] = 0;
                    dataPointsDIPb[3] =
user.getMeasurementsLittle().get(0).getDipAngleBent();
                } else {
                    dataPointsDIPb[i * 2 + 1] =
user.getMeasurementsLittle().get(i - 1).getDipAngleBent();
                }
            }
        }
        break;

        case "RINGFINGER":

            dataPointsDIPb = new
Number[(user.getMeasurementsRing().size() * 2) + 2];

            for (int i = 1; i < user.getMeasurementsRing().size() + 1;
i++) {
                if (i == 1) {
                    dataPointsDIPb[0] = 0;
                    dataPointsDIPb[2] = 1;
                } else {
                    dataPointsDIPb[i * 2] = i;
                }
            }
    }
}

```

```

        }

        for (int i = 1; i < (user.getMeasurementsRing().size() +
1); i++) {
            if (i == 1) {
                dataPointsDIPb[1] = 0;
                dataPointsDIPb[3] =
user.getMeasurementsRing().get(0).getDipAngleBent();
            } else {
                dataPointsDIPb[i * 2 + 1] =
user.getMeasurementsRing().get(i - 1).getDipAngleBent();
            }
        }
        break;

    case "LÅNGFINGER":

        dataPointsDIPb = new
Number[(user.getMeasurementsMiddle().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsMiddle().size() +
1; i++) {
            if (i == 1) {
                dataPointsDIPb[0] = 0;
                dataPointsDIPb[2] = 1;
            } else {
                dataPointsDIPb[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsMiddle().size() +
1); i++) {
            if (i == 1) {
                dataPointsDIPb[1] = 0;
                dataPointsDIPb[3] =
user.getMeasurementsMiddle().get(0).getDipAngleBent();
            } else {
                dataPointsDIPb[i * 2 + 1] =
user.getMeasurementsMiddle().get(i - 1).getDipAngleBent();
            }
        }
        break;

    case "PEKFINGER":

        dataPointsDIPb = new
Number[(user.getMeasurementsIndex().size() * 2) + 2];

        for (int i = 1; i < user.getMeasurementsIndex().size() + 1;
i++) {
            if (i == 1) {
                dataPointsDIPb[0] = 0;
                dataPointsDIPb[2] = 1;
            } else {
                dataPointsDIPb[i * 2] = i;
            }
        }

        for (int i = 1; i < (user.getMeasurementsIndex().size() +
1); i++) {
            if (i == 1) {

```

```

        dataPointsDIPb[1] =
0;//user.getMeasurements().get(0).getTiltValue();
        dataPointsDIPb[3] =
user.getMeasurementsIndex().get(0).getDipAngleBent();
    } else {
        dataPointsDIPb[i * 2 + 1] =
user.getMeasurementsIndex().get(i - 1).getDipAngleBent();
    }
}
break;
}
return dataPointsDIPb;
}

@Override
public void onBackPressed() {
    Intent onBackIntent = new Intent(HistoryActivity.this,
 BaseActivity.class);
    startActivity(onBackIntent);
    finish();
}

@Override
protected void onPause() {
    saveSharedPreferencesHistory();
    super.onPause();
}

@Override
protected void onResume() {
    super.onResume();
    checkSharedPreferencesHistory();
}

private void saveSharedPreferencesHistory() {
    String jsonString = gson.toJson(user);
    editor.putString("com.example.digitalgoniometer.user",
jsonString).commit();
    String jsonStringHistory = gson.toJson(historyInfo);
    editor.putString("com.example.digitalgoniometer.historyInfo",
jsonStringHistory).commit();
}

protected void checkSharedPreferencesHistory() {
    String receivedString =
sharedPreferences.getString("com.example.digitalgoniometer.user", null);
    if (receivedString != null && !receivedString.equals("null")) {
        user = gson.fromJson(receivedString, User.class);
    }
    String receivedStringHistory =
sharedPreferences.getString("com.example.digitalgoniometer.historyInfo",
null);
    if (receivedStringHistory != null &&
!receivedStringHistory.equals("null")) {
        historyInfo = gson.fromJson(receivedStringHistory,
HistoryInfo.class);
    }
}
}

```

MeasureActivity:

```
package com.example.digitalgoniometer;

import com.example.digitalgoniometer.model.Dot;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AlertDialog;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
import android.content.DialogInterface;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.media.ExifInterface;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Environment;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.view.ViewTreeObserver;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;
import android.Manifest;
import android.widget.VideoView;
import com.example.digitalgoniometer.model.MeasurementInfo;
import com.example.digitalgoniometer.views.CustomizedImageView;
import java.io.File;
import java.io.IOException;
import java.util.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
```

```
public class MeasureActivity extends BaseActivity {

    private static final int REQUEST_CODE_CAMERA = 1001;
    private File imageFile = null;
    private String imagePath = null;
    private CustomizedImageView customizedImageView;
    private static ImageView zoomImageView;
    private ImageView infoImageView;
    private Bitmap imageBitmap;
    private Button saveButton, takeNewPictureButton, cancelButton,
infoButton;
    private TextView infoTextView;
    private double[] anglesBent;
    private double[] anglesStraight;
    private int currentFingerMeasure;
    private boolean originalWantInfo, wantInfo, wantInfo2,
firstCameraStarted, secondCameraStarted;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_measure);

    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        actionBar.hide();
    }

    checkSharedPreferences();

    customizedImageView = findViewById(R.id.customizedImageView);
    saveButton = findViewById(R.id.saveButton);
    takeNewPictureButton = findViewById(R.id.takeNewPictureButton);
    infoButton = findViewById(R.id.infoButton);
    cancelButton = findViewById(R.id.cancelMeasureButton);
    infoImageView = findViewById(R.id.infoImageView);
    zoomImageView = findViewById(R.id.zoomImageView);
    infoTextView = findViewById(R.id.infoTextMeasureView);

    anglesBent = null;
    anglesStraight = null;

    Intent fromIntent = getIntent();
    originalWantInfo = fromIntent.getBooleanExtra("WantInfo", true);
    wantInfo = originalWantInfo;
    wantInfo2 = false;
    firstCameraStarted = false;
    secondCameraStarted = false;

    setFingerInfoImage();
    updateInfoText();

    saveButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            if (anglesStraight == null) {
                anglesStraight = calculateAngles();

                if (wantInfo2) {
                    giveThirdInfoPopup();
                } else {
                    startCamera();
                }
            }

            if (anglesBent == null) {
                anglesBent = calculateAngles();
                saveData();
            }
        }
    });

    takeNewPictureButton.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        startCamera();
}
});

```

```

        }
    });

cancelButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent cancelIntent = new Intent(MeasureActivity.this,
BaseActivity.class);
        startActivity(cancelIntent);
        finish();
    }
});

infoButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (anglesStraight==null){
            View measureInfoSecondDialogView =
getLayoutInflater().inflate(R.layout.measure_info_second_dialog, null);
            AlertDialog.Builder secondInfoAlertDialogBuilder = new
AlertDialog.Builder(MeasureActivity.this).setView(measureInfoSecondDialogVi
ew).setTitle("BILD ETT: STRÄCKTA FINGRAR");

            VideoView infoVideoView =
measureInfoSecondDialogView.findViewById(R.id.secondInfoVideo);
            TextView secondHeader =
measureInfoSecondDialogView.findViewById(R.id.SecondHeader);
            TextView secondMainText =
measureInfoSecondDialogView.findViewById(R.id.SecondMainText);
            secondHeader.setText("Du ska nu mäta ditt "+user.getFingersAffected().get(currentFingerMeasure).toLowerCase());
            secondMainText.setText("Du kommer nu få ta en bild på
din skadade hand med fingrarna så sträckta som möjligt.\n\nTa bilden så att
ditt "+ user.getFingersAffected().get(currentFingerMeasure).toLowerCase()+" syns väl med kameran rakt framför handens sida.\n\nDu ska därefter
placera ut punkter på dina ledar på det skadade fingret.\n\nFlytta
punkterna genom att dra dem på skärmen till rätt led. \n\nNär du är nöjd
med placeringen klickar du på spara.");
            String videoPath = null;

            switch
(user.getFingersAffected().get(currentFingerMeasure)) {

                case "PEKFINGER":
                    videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_index;
                    break;

                case "LÅNGFINGER":
                    videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_middle;
                    break;

                case "RINGFINGER":
                    videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_ring;
                    break;

                case "LILLFINGER":
                    videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_little;
                    break;
            }
        }
    }
});

```

```

        break;

    }

    MediaController mediaController = new
MediaController(MeasureActivity.this);
    mediaController.setAnchorView(infoVideoView);
    infoVideoView.setMediaController(mediaController);

    Uri videoUri = Uri.parse(videoPath);
    infoVideoView.setVideoURI(videoUri);

    VideoView finalInfoVideoView = infoVideoView;
    infoVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {
        finalInfoVideoView.start();
    }
});

secondInfoAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        }
    });

secondInfoAlertBuilder.create();
secondInfoAlertBuilder.show();
} else if (anglesBent==null){
    View measureInfoThirdDialogView =
getLayoutInflater().inflate(R.layout.measure_info_third_dialog, null);
    AlertDialog.Builder thirdInfoAlertDialog = new
AlertDialog.Builder(MeasureActivity.this).setView(measureInfoThirdDialogView)
    .setTitle("BILD TVÅ: BÖJDA FINGRAR");

    VideoView infoVideoView =
measureInfoThirdDialogView.findViewById(R.id.thirdInfoVideo);
    TextView secondHeader =
measureInfoThirdDialogView.findViewById(R.id.thirdHeader);
    TextView secondMainText =
measureInfoThirdDialogView.findViewById(R.id.thirdMainText);
    secondHeader.setText("Du ska nu mäta ditt "+user.getFingersAffected().get(currentFingerMeasure).toLowerCase());
    secondMainText.setText("Du kommer nu få ta en bild på din skadade hand med fingrarna så böjda som möjligt.\n\nTa bilden så ditt "+user.getFingersAffected().get(currentFingerMeasure).toLowerCase()+" syns väl med kameran rakt framför handens sida.\n\nDu ska därefter placera ut punkter på dina leder på det skadade fingret.\n\nFlytta punkterna genom att dra dem på skärmen till rätt led. \n\nNär du är nöjd med placeringen klickar du på spara.");
}

String videoPath = null;

switch
(user.getFingersAffected().get(currentFingerMeasure)) {

    case "PEKFINGER":
```

```

                videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.bent_index;
                break;

            case "LÅNGFINGER":
                videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.bent_middle;
                break;

            case "RINGFINGER":
                videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.bent_ring;
                break;

            case "LILLFINGER":
                videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.bent_little;
                break;

        }

        MediaController mediaController = new
MediaController(MeasureActivity.this);
        mediaController.setAnchorView(infoVideoView);
        infoVideoView.setMediaController(mediaController);

        Uri videoUri = Uri.parse(videoPath);
        infoVideoView.setVideoURI(videoUri);

        VideoView finalInfoVideoView = infoVideoView;
        infoVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
            @Override
            public void onPrepared(MediaPlayer mp) {
                finalInfoVideoView.start();
            }
        });

        thirdInfoAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                }
            });
        thirdInfoAlertBuilder.create();
        thirdInfoAlertBuilder.show();
    }
}
});

currentFingerMeasure = 0;
startMeasure();

}

private void setFingerInfoImage() {
    if (anglesStraight==null){

```

```

        if (user.isRightHand()) {
            switch
(user.getFingersAffected().get(currentFingerMeasure)) {
                case "PEKFINGER":

infoImageView.setImageResource(R.drawable.index_straight_h);
                break;

                case "LÅNGFINGER":

infoImageView.setImageResource(R.drawable.middle_straight_h);
                break;

                case "RINGFINGER":

infoImageView.setImageResource(R.drawable.ring_straight_h);
                break;

                case "LILLFINGER":

infoImageView.setImageResource(R.drawable.little_straight_h);
                break;
            } else {
                switch
(user.getFingersAffected().get(currentFingerMeasure)) {
                    case "PEKFINGER":

infoImageView.setImageResource(R.drawable.index_straight_v);
                break;

                    case "LÅNGFINGER":

infoImageView.setImageResource(R.drawable.middle_straight_v);
                break;

                    case "RINGFINGER":

infoImageView.setImageResource(R.drawable.ring_straight_v);
                break;

                    case "LILLFINGER":

infoImageView.setImageResource(R.drawable.little_straight_v);
                break;
            }
        } else if (anglesBent==null){
            if (user.isRightHand()) {
                switch
(user.getFingersAffected().get(currentFingerMeasure)) {
                    case "PEKFINGER":

infoImageView.setImageResource(R.drawable.index_bent_h);
                break;

                    case "LÅNGFINGER":

infoImageView.setImageResource(R.drawable.middle_bent_h);
                break;

                    case "RINGFINGER":

```

```

infoImageView.setImageResource(R.drawable.ring_bent_h);
        break;

    case "LILLFINGER":

infoImageView.setImageResource(R.drawable.little_bent_h);
        break;
    } else {
        switch
(user.getFingersAffected().get(currentFingerMeasure)) {
            case "PEKFINGER":

infoImageView.setImageResource(R.drawable.index_bent_v);
        break;

        case "LÄNGFINGER":

infoImageView.setImageResource(R.drawable.middle_bent_v);
        break;

        case "RINGFINGER":

infoImageView.setImageResource(R.drawable.ring_bent_v);
        break;

        case "LILLFINGER":

infoImageView.setImageResource(R.drawable.little_bent_v);
        break;
    }
}

private void updateInfoText(){
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("Mät ditt");
    if (user.isRightHand()){
        stringBuilder.append(" högra ");
    } else if (!user.isRightHand()) {
        stringBuilder.append(" vänstra ");
    }
    switch (user.getFingersAffected().get(currentFingerMeasure)) {

        case "PEKFINGER":
            stringBuilder.append("pekfinger ");
            break;

        case "LÄNGFINGER":
            stringBuilder.append("långfinger ");
            break;

        case "RINGFINGER":
            stringBuilder.append("ringfinger ");
            break;

        case "LILLFINGER":
            stringBuilder.append("lillfinger ");
            break;
    }
}

```

```

        }

        if (anglesStraight==null){
            stringBuilder.append("så sträckt som möjligt");
        } else if (anglesBent==null) {
            stringBuilder.append("så böjt som möjligt");
        }
        infoTextView.setText(stringBuilder.toString());
    }

    private void startMeasure() {
        if (anglesStraight == null){
            giveSecondInfoPopup();
        }
    }

    private void giveSecondInfoPopup() {
        if (wantInfo) {
            if (!firstCameraStarted){
                firstCameraStarted = true;
                View measureInfoSecondDialogView =
getLayoutInflater().inflate(R.layout.measure_info_second_dialog, null);
                AlertDialog.Builder secondInfoAlertDialogBuilder = new
AlertDialog.Builder(MeasureActivity.this).setView(measureInfoSecondDialogVi
ew).setTitle("BILD ETT: STRÄCKTA FINGRAR");

                VideoView infoVideoView =
measureInfoSecondDialogView.findViewById(R.id.secondInfoVideo);
                TextView secondHeader =
measureInfoSecondDialogView.findViewById(R.id.SecondHeader);
                TextView secondMainText =
measureInfoSecondDialogView.findViewById(R.id.SecondMainText);
                ScrollView secondScrollView =
measureInfoSecondDialogView.findViewById(R.id.scrollViewSecond);
                secondScrollView.scrollTo(0, 0);
                secondHeader.setText("Du ska nu mäta ditt "+

user.getFingersAffected().get(currentFingerMeasure).toLowerCase()+" med
fingrarna så sträckta som möjligt.");
                secondMainText.setText("\nTa bilden så att ditt "+
user.getFingersAffected().get(currentFingerMeasure).toLowerCase() +" syns
väl med kameran rakt framför handens sida.\n\nDu ska därefter placera ut
punkter på dina leder på det skadade fingret.\n\nPlacera punkterna längs
konturen av fingret längs ovansidan, som videon visar. \n\nFlytta punkterna
genom att dra dem på skärmen till rätt led. \n\nNär du är nöjd med
placeringen klickar du på spara.");
                String videoPath = null;

                switch
(user.getFingersAffected().get(currentFingerMeasure)) {

                    case "PEKFINGER":
                        videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_index;
                        break;

                    case "LÅNGFINGER":
                        videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_middle;
                        break;
                }
            }
        }
    }
}

```

```

        case "RINGFINGER":
            videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_ring;
            break;

        case "LILLFINGER":
            videoPath = "android.resource://" +
getPackageName() + "/" + R.raw.straight_little;
            break;

    }

    MediaController mediaController = new
MediaController(this);
    mediaController.setAnchorView(infoVideoView);
    infoVideoView.setMediaController(mediaController);

    Uri videoUri = Uri.parse(videoPath);
    infoVideoView.setVideoURI(videoUri);

    VideoView finalInfoVideoView = infoVideoView;
    infoVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {
        finalInfoVideoView.start();
    }
});

    secondInfoAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface,
int i) {
        wantInfo = false;
        wantInfo2 = true;
        firstCameraStarted = true;
        setFingerInfoImage();
        updateInfoText();
        startCamera();

    }
});
    secondInfoAlertBuilder.setNegativeButton("Avbryt", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface,
int i) {
        Intent goBackActivityIntent = new
Intent(MeasureActivity.this, BaseActivity.class);
        startActivity(goBackActivityIntent);
        finish();

    }
});

    secondInfoAlertBuilder.setOnDismissListener(new
DialogInterface.OnDismissListener() {
    @Override
    public void onDismiss(DialogInterface dialog) {
        if (!firstCameraStarted) {

```

```

                giveSecondInfoPopup();
            }
        }
    });
    secondInfoAlertDialogBuilder.create();
    secondInfoAlertDialogBuilder.show();
} else{

}

} else{
    startCamera();
}
}

private void giveThirdInfoPopup() {
    secondCameraStarted = false;
    View measureInfoThirdDialogView =
getLayoutInflater().inflate(R.layout.measure_info_third_dialog, null);
    AlertDialog.Builder thirdInfoAlertDialogBuilder = new
AlertDialog.Builder(MeasureActivity.this).setView(measureInfoThirdDialogView)
    .setTitle("BILD TVÅ: BÖJDA FINGRAR");

    VideoView infoVideoView =
measureInfoThirdDialogView.findViewById(R.id.thirdInfoVideo);
    TextView secondHeader =
measureInfoThirdDialogView.findViewById(R.id.thirdHeader);
    TextView secondMainText =
measureInfoThirdDialogView.findViewById(R.id.thirdMainText);
    ScrollView thirdScrollView =
measureInfoThirdDialogView.findViewById(R.id.scrollViewThird);
    thirdScrollView.scrollTo(0, 0);
    secondHeader.setText("Du ska nu mäta ditt "+  

user.getFingersAffected().get(currentFingerMeasure).toLowerCase() + " med  

fingrarna så böjda som möjligt.");
    secondMainText.setText("\nTa bilden så ditt "+  

user.getFingersAffected().get(currentFingerMeasure).toLowerCase() +" syns  

väl med kameran rakt framför handens sida.\n\nDu ska därefter placera ut  

punkter på dina leder på det skadade fingret.\n\nPlacera punkterna längs  

konturen av fingret längs ovansidan, som videon visar. \n\nFlytta punkterna  

genom att dra dem på skärmen till rätt led. \n\nNär du är nöjd med  

placeringen klickar du på spara.");
}

String videoPath = null;

switch (user.getFingersAffected().get(currentFingerMeasure)) {

    case "PEKFINGER":
        videoPath = "android.resource://" + getPackageName() + "/"  

+ R.raw.bent_index;
        break;

    case "LÄNGFINGER":
        videoPath = "android.resource://" + getPackageName() + "/"  

+ R.raw.bent_middle;
        break;

    case "RINGFINGER":
        videoPath = "android.resource://" + getPackageName() + "/"  

+ R.raw.bent_ring;
        break;
}

```

```

        case "LILLFINGER":
            videoPath = "android.resource://" + getPackageName() + "/"
+ R.raw.bent_little;
            break;

    }

    MediaController mediaController = new MediaController(this);
    mediaController.setAnchorView(infoVideoView);
    infoVideoView.setMediaController(mediaController);

    Uri videoUri = Uri.parse(videoPath);
    infoVideoView.setVideoURI(videoUri);

    VideoView finalInfoVideoView = infoVideoView;
    infoVideoView.setOnPreparedListener(new
MediaPlayer.OnPreparedListener() {
        @Override
        public void onPrepared(MediaPlayer mp) {
            finalInfoVideoView.start();
        }
    });
}

thirdInfoAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        secondCameraStarted = true;
        setFingerInfoImage();
        updateInfoText();
        startCamera();
    }
});

thirdInfoAlertBuilder.create();
thirdInfoAlertBuilder.setOnDismissListener(new
DialogInterface.OnDismissListener() {
    @Override
    public void onDismiss(DialogInterface dialog) {
        if (!secondCameraStarted){
            giveThirdInfoPopup();
        }
    }
});
thirdInfoAlertBuilder.show();
}

public double[] calculateAngles() {

    double[] tempArray = new double[3];
    ArrayList<Dot> dotCoordinates =
customizedImageView.getCoordinates();

    double xTip, yTip, xDip, yDip, xPip, yPip, xMcp, yMcp, xHand,
yHand;
    xTip = dotCoordinates.get(dotCoordinates.size() - 1).getX();
    yTip = dotCoordinates.get(dotCoordinates.size() - 1).getY();
    xDip = dotCoordinates.get(dotCoordinates.size() - 2).getX();
    yDip = dotCoordinates.get(dotCoordinates.size() - 2).getY();
    xPip = dotCoordinates.get(dotCoordinates.size() - 3).getX();
}

```

```

        yPip = dotCoordinates.get(dotCoordinates.size() - 3).getY();
        xMcp = dotCoordinates.get(dotCoordinates.size() - 4).getX();
        yMcp = dotCoordinates.get(dotCoordinates.size() - 4).getY();
        xHand = dotCoordinates.get(dotCoordinates.size() - 5).getX();
        yHand = dotCoordinates.get(dotCoordinates.size() - 5).getY();

        double a = Math.sqrt(Math.pow((xTip - xPip), 2) + (Math.pow((yTip - yPip), 2))); // a = length tip-pip
        double b = Math.sqrt(Math.pow((xDip - xPip), 2) + (Math.pow((yDip - yPip), 2))); // b = length dip-pip
        double c = Math.sqrt(Math.pow((xDip - xTip), 2) + (Math.pow((yDip - yTip), 2))); // c = length dip-tip
        double d = Math.sqrt(Math.pow((xPip - xMcp), 2) + (Math.pow((yPip - yMcp), 2))); // d = length pip-mcp
        double e = Math.sqrt(Math.pow((xDip - xMcp), 2) + (Math.pow((yDip - yMcp), 2))); // e = length dip-mcp
        double f = Math.sqrt(Math.pow((xMcp - xHand), 2) + (Math.pow((yMcp - yHand), 2))); // f = length mcp-hand
        double g = Math.sqrt(Math.pow((xPip - xHand), 2) + (Math.pow((yPip - yHand), 2))); // g = length pip-mcp

        double dipAngle = 180 - (Math.toDegrees(Math.acos((Math.pow(a, 2) - Math.pow(b, 2) - Math.pow(c, 2)) / (-2 * b * c)))) ;
        double pipAngle = 180 - (Math.toDegrees(Math.acos((Math.pow(e, 2) - Math.pow(d, 2) - Math.pow(b, 2)) / (-2 * b * d)))) ;
        double mcpAngle = 180 - (Math.toDegrees(Math.acos((Math.pow(g, 2) - Math.pow(d, 2) - Math.pow(f, 2)) / (-2 * d * f)))) ;

        tempArray[0] = dipAngle;
        tempArray[1] = pipAngle;
        tempArray[2] = mcpAngle;

        return tempArray;
    }

    private void startCamera(){
        if (ContextCompat.checkSelfPermission(getApplicationContext(), Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
            getImage();
        } else {
            requestPermissionLauncher.launch(Manifest.permission.CAMERA);
        }
    }

    private void getImage() {

        Intent getImageIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (getImageIntent.resolveActivity(getPackageManager()) != null) {
            try {
                createImageFile();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }

            if (imageFile != null) {
                Uri imageURI =
FileProvider.getUriForFile(MeasureActivity.this,
"com.example.digitalgoniometer.file_provider_paths", imageFile);
                getImageIntent.putExtra(MediaStore.EXTRA_OUTPUT, imageURI);
            }
        }
    }
}

```

```

        StringBuider stringBuilder = new StringBuider();
        if (anglesStraight==null){
            stringBuilder.append("Mät rakt ");
        } else if (anglesBent == null){
            stringBuilder.append("Mät böjt ");
        }

        switch
(user.getFingersAffected().get(currentFingerMeasure)) {

    case "PEKFINGER":
        stringBuilder.append("pekfinger");
        break;

    case "LÅNGFINGER":
        stringBuilder.append("långfinger");
        break;

    case "RINGFINGER":
        stringBuilder.append("ringfinger");
        break;

    case "LILLFINGER":
        stringBuilder.append("lillfinger");
        break;
}

        Toast toast = Toast.makeText(MeasureActivity.this,
stringBuilder.toString(), Toast.LENGTH_LONG);
        toast.show();

        startActivityForResult(getImageIntent,
REQUEST_CODE_CAMERA);

    }

}

@Override
protected void onActivityResult ( int requestCode, int resultCode,
Intent data){
    super.onActivityResult(requestCode, resultCode, data); //DENNA6

    if (resultCode == 0){
        backToMain();
    } else if (requestCode == REQUEST_CODE_CAMERA && resultCode ==
RESULT_OK) {

        customizedImageView.setVisibility(View.VISIBLE);

        ViewTreeObserver viewTreeObserver =
customizedImageView.getViewTreeObserver();
        viewTreeObserver.addOnGlobalLayoutListener(new
ViewTreeObserver.OnGlobalLayoutListener() {
            @Override
            public void onGlobalLayout() {

customizedImageView.getViewTreeObserver().removeOnGlobalLayoutListener(this
);

```

```

        customizedImageView.resetCoordinates();
        Bitmap bitmap =
BitmapFactory.decodeFile(imageFilePath);
        bitmap = getCorrectOrientation(imageFilePath, bitmap);

        int bitmapWidth = bitmap.getWidth();
        int bitmapHeight = bitmap.getHeight();
        int imageViewWidth = customizedImageView.getWidth();
        int imageViewHeight = customizedImageView.getHeight();

        float scaleFactor = Math.max((float) imageViewWidth /
bitmapWidth, (float) imageViewHeight / bitmapHeight);

        int scaledWidth = Math.round(bitmapWidth *
scaleFactor);
        int scaledHeight = Math.round(bitmapHeight *
scaleFactor);

        Bitmap scaledBitmap = Bitmap.createScaledBitmap(bitmap,
scaledWidth, scaledHeight, true);

        imageBitmap=scaledBitmap;

        if (scaledBitmap.getWidth()>imageViewWidth){
            imageBitmap = Bitmap.createBitmap(scaledBitmap,
((scaledBitmap.getWidth()-imageViewWidth)/2), 0,imageViewWidth,
scaledBitmap.getHeight());
        }
        if (scaledBitmap.getHeight()>imageViewHeight){
            imageBitmap = Bitmap.createBitmap(scaledBitmap, 0,
((scaledBitmap.getHeight()-imageViewHeight)/2),scaledBitmap.getWidth(),
imageViewHeight);
        }

        customizedImageView.setImageBitmap(imageBitmap);

customizedImageView.setDotsOnScreen(user.isRightHand());
updateInfoText();
setFingerInfoImage();
    }

}

private void createImageFile () throws IOException{
    String imageFileName = "Image";
    File externalStorageDir =
getExternalFilesDir(Environment.DIRECTORY_PICTURES);

    if (imageFile == null) {
        imageFile = File.createTempFile(imageFileName, ".jpg",
externalStorageDir);
    }
    imagePath = imageFile.getAbsolutePath();
}

public static Bitmap rotateImage(Bitmap source, float angle) {

```

```

        Matrix matrix = new Matrix();
        matrix.postRotate(angle);
        return Bitmap.createBitmap(source, 0, 0, source.getWidth(),
source.getHeight(),
                matrix, true);
    }

    private Bitmap getCorrectOrientation(String imagePath, Bitmap
bitmap) {
    try {
        ExifInterface exifInterface = new ExifInterface(imagePath);
        int orientation =
exifInterface.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_UNDEFINED);
        switch (orientation) {
            case ExifInterface.ORIENTATION_ROTATE_90:
                return rotateImage(bitmap, 90);

            case ExifInterface.ORIENTATION_ROTATE_180:
                return rotateImage(bitmap, 180);

            case ExifInterface.ORIENTATION_ROTATE_270:
                return rotateImage(bitmap, 270);

            case ExifInterface.ORIENTATION_NORMAL:
            default:
                return bitmap;
        }
    } catch (IOException ex) {
        ex.printStackTrace();
        return null;
    }
}
private ActivityResultLauncher<String> requestPermissionLauncher =
registerForActivityResult(new
ActivityResultContracts.RequestPermission(), isGranted -> {
    if (isGranted) {
        Log.d("permission", "permission granted");
        getImage();
    } else {
        Log.d("permission", "permission not granted");
    }
});

public static void setZoomedBitmap(Bitmap bitmap) {
    zoomImageView.setImageBitmap(bitmap);
}

public static void resetZoomedBitmap() {
    zoomImageView.setImageBitmap(null);
}

private void saveData() {
    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
    Date today = new Date();
    String dateString = dateFormat.format(today);

    if (anglesBent == null || anglesStraight == null){
        Toast toast = Toast.makeText(MeasureActivity.this, "Gör båda
mätningarna", Toast.LENGTH_SHORT);
        toast.show();
    }
}

```

```

        } else {
            checkMeasurementExists(dateString);
        }
    }

private void checkMeasurementExists(String dateString) {

    switch (user.getFingersAffected().get(currentFingerMeasure)) {
        case "PEKFINGER":if (user.getMeasurementsIndex().size() == 0 || !user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 1).getDate().equals(dateString)) {
            saveDataDialog();
        } else if (user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 1).getDate().equals(dateString)) {
            overWriteDataDialog();
        }break;

        case "LÅNGFINGER": if (user.getMeasurementsMiddle().size() == 0 || !user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 1).getDate().equals(dateString)) {
            saveDataDialog();
        } else if (user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 1).getDate().equals(dateString)) {
            overWriteDataDialog();
        }break;

        case "RINGFINGER":if (user.getMeasurementsRing().size() == 0 || !user.getMeasurementsRing().get(user.getMeasurementsRing().size() - 1).getDate().equals(dateString)) {
            saveDataDialog();
        } else if (user.getMeasurementsRing().get(user.getMeasurementsRing().size() - 1).getDate().equals(dateString)) {
            overWriteDataDialog();
        }break;

        case "LILLFINGER":
            if (user.getMeasurementsLittle().size() == 0 || !user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() - 1).getDate().equals(dateString)) {
                saveDataDialog();
            } else if (user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() - 1).getDate().equals(dateString)) {
                overWriteDataDialog();
            }break;
    }
}

private void overWriteDataDialog() {
    AlertDialog.Builder builderOverWriteDataDialog = new AlertDialog.Builder(MeasureActivity.this);
    builderOverWriteDataDialog.setTitle("Du har redan gjort en mätning idag, vill du ersätta den med följande data?");
    builderOverWriteDataDialog.setMessage("Sträckta fingrar: DIP - "+(int) anglesStraight[0] + ", PIP - " + (int) anglesStraight[1] + "\nBöjda fingrar: DIP - "+ (int) anglesBent[0] + ", PIP - " + (int) anglesBent[1]);
}

```

```

        final boolean[] hasChoosen = {false};
        builderOverWriteDataDialog.setPositiveButton("JA", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                Date today = new Date();
                String dateString = dateFormat.format(today);
                changeLast(dateString);
                hasChoosen[0] = true;
                giveWarningsIfFault();
            }
        });
        builderOverWriteDataDialog.setNegativeButton("NEJ", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                anglesBent = null;
                anglesStraight = null;
                hasChoosen[0] = true;
                backToMain();
            }
        });

        builderOverWriteDataDialog.create();
        builderOverWriteDataDialog.setOnDismissListener(new
DialogInterface.OnDismissListener() {
            @Override
            public void onDismiss(DialogInterface dialog) {
                if (!hasChoosen[0]){
                    overWriteDataDialog();
                }
            }
        });
        builderOverWriteDataDialog.show();
    }

    private void changeLast(String dateString) {
        switch (user.getFingersAffected().get(currentFingerMeasure)) {

            case "PEKFINGER":
                user.changeLastMeasurementIndex(new
MeasurementInfo(dateString, (int) anglesBent[1], (int) anglesStraight[1],
(int) anglesBent[0], (int) anglesStraight[0], (int) anglesBent[2], (int)
anglesStraight[2]));
                break;

            case "LÅNGFINGER":
                user.changeLastMeasurementMiddle(new
MeasurementInfo(dateString, (int) anglesBent[1], (int) anglesStraight[1],
(int) anglesBent[0], (int) anglesStraight[0], (int) anglesBent[2], (int)
anglesStraight[2]));
                break;

            case "RINGFINGER":
                user.changeLastMeasurementRing(new
MeasurementInfo(dateString, (int) anglesBent[1], (int) anglesStraight[1],
(int) anglesBent[0], (int) anglesStraight[0], (int) anglesBent[2], (int)
anglesStraight[2]));
                break;

            case "LILLFINGER":
```

```

        user.changeLastMeasurementLittle(new
MeasurementInfo(dateString, (int) anglesBent[1], (int) anglesStraight[1],
(int) anglesBent[0], (int) anglesStraight[0], (int) anglesBent[2], (int)
anglesStraight[2]));
        break;
    }
}

private void saveDataDialog() {
    AlertDialog.Builder builderSaveDataDialog = new
AlertDialog.Builder(MeasureActivity.this);
    builderSaveDataDialog.setTitle("Vill du spara mätningen?");
    int textTAM = (int) ((anglesBent[1] - anglesStraight[1]) +
(anglesBent[0] - anglesStraight[0]));
    builderSaveDataDialog.setMessage("Sträckta fingrar: \nDIP " + (int)
anglesStraight[0] + "°" + ", PIP " + (int) anglesStraight[1] + "°"+ ", MCP
" + (int) anglesStraight[2] + "°"
        + "\nBöjda fingrar: \nDIP " + (int) anglesBent[0] + "°" + ",
PIP " + (int) anglesBent[1]+ "°" + ", MCP " + (int) anglesBent[2]+ "°\n"
+ "TAM: " + textTAM );

    final boolean[] hasChoosen = {false};
    builderSaveDataDialog.setPositiveButton("JA", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
            Date today = new Date();
            String dateString = dateFormat.format(today);
            addToFinger(dateString);
            hasChoosen[0] = true;
            giveWarningsIfFault();
        }
    });
    builderSaveDataDialog.setNegativeButton("NEJ", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            anglesBent = null;
            anglesStraight = null;
            hasChoosen[0] = true;
            backToMain();
            //TODO vill vi till main?
        }
    });
    builderSaveDataDialog.create();
    builderSaveDataDialog.setOnDismissListener(new
DialogInterface.OnDismissListener() {
        @Override
        public void onDismiss(DialogInterface dialog) {
            if (!hasChoosen[0]){
                saveDataDialog();
            }
        }
    });
}

builderSaveDataDialog.show();
}

private void giveWarningsIfFault() {
    AlertDialog.Builder builderWarningDialog = new

```

```

AlertDialog.Builder(MeasureActivity.this);
builderWarningDialog.setTitle("Vill du spara mätningen?");
boolean warningActivated = false;

if (((anglesBent[1] - anglesStraight[1]) + (anglesBent[0] -
anglesStraight[0])) < 0) {
    warningActivated = true;
    builderWarningDialog.setMessage("TAM har ett negativt värde,
dubbelkolla att du tog bilderna i rätt ordning.\nVill du spara
ändå?"); //TODO dissutera med handledare om vad det ska stå
}

switch (user.getFingersAffected().get(currentFingerMeasure)) {

    case "PEKFINGER":
        if (user.getMeasurementsIndex().size() > 1) {
            if ((Math.abs(anglesBent[0] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getDipAngleBent())) > 10) ||
                (Math.abs(anglesStraight[0] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getDipAngleStraight())) > 10) ||
                (Math.abs(anglesBent[1] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getPipAngleBent())) > 10) ||
                (Math.abs(anglesStraight[1] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getPipAngleStraight())) > 10) ||
                (Math.abs(anglesBent[2] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getMcpAngleBent())) > 10) ||
                (Math.abs(anglesStraight[2] -
(user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getMcpAngleStraight())) > 10) ||
                Math.abs((anglesBent[1] - anglesStraight[1]) +
(anglesBent[0] - anglesStraight[0]) -
user.getMeasurementsIndex().get(user.getMeasurementsIndex().size() - 2).getTamValue()) > 20) {
                    warningActivated = true;
                    builderWarningDialog.setMessage("Något av dina värden
avviker mycket från din förra mätning.\nVill du spara ändå?");
                }
            }
        }
        break;

    case "LÅNGFINGER":
        if (user.getMeasurementsMiddle().size() > 1) {
            if ((Math.abs(anglesBent[0] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getDipAngleBent())) > 10) ||
                (Math.abs(anglesStraight[0] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getDipAngleStraight())) > 10) ||
                (Math.abs(anglesBent[1] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getPipAngleBent())) > 10) ||
                (Math.abs(anglesStraight[1] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getPipAngleStraight())) > 10) ||
                (Math.abs(anglesBent[2] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getMcpAngleBent())) > 10) ||
                (Math.abs(anglesStraight[2] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() - 2).getMcpAngleStraight())) > 10))
            )
        }
}

```

```

2).getMcpAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[2] -
(user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() -
2).getMcpAngleStraight())) > 10) ||
                    Math.abs((anglesBent[1] - anglesStraight[1]) +
(anglesBent[0] - anglesStraight[0]) -
user.getMeasurementsMiddle().get(user.getMeasurementsMiddle().size() -
2).getTamValue()) > 20) {
                warningActivated = true;
                builderWarningDialog.setMessage("Något av dina
värden avviker mycket från din förra mätning.\nVill du spara ändå?");
            }
        }
        break;

    case "RINGFINGER":
        if (user.getMeasurementsRing().size()>1) {
            if ((Math.abs(anglesBent[0] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getDipAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[0] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getDipAngleStraight())) > 10) ||
                    (Math.abs(anglesBent[1] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getPipAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[1] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getPipAngleStraight())) > 10) ||
                    (Math.abs(anglesBent[2] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getMcpAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[2] -
(user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getMcpAngleStraight())) > 10) ||
                    Math.abs((anglesBent[1] - anglesStraight[1]) +
(anglesBent[0] - anglesStraight[0]) -
user.getMeasurementsRing().get(user.getMeasurementsRing().size() -
2).getTamValue()) > 20) {
                warningActivated = true;
                builderWarningDialog.setMessage("Något av dina
värden avviker mycket från din förra mätning.\nVill du spara ändå?");
            }
        }
        break;

    case "LILLFINGER":
        if (user.getMeasurementsLittle().size()>1) {
            if ((Math.abs(anglesBent[0] -
(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() -
2).getDipAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[0] -
(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() -
2).getDipAngleStraight())) > 10) ||
                    (Math.abs(anglesBent[1] -
(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() -
2).getPipAngleBent())) > 10) ||
                    (Math.abs(anglesStraight[1] -
(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() -
2).getPipAngleStraight())) > 10) ||
                    (Math.abs(anglesBent[2] -

```

```

(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() - 2).getMcpAngleBent())) > 10) ||
(Math.abs(anglesStraight[2] - (user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() - 2).getMcpAngleStraight())) > 10) ||
Math.abs((anglesBent[1] - anglesStraight[1]) + (anglesBent[0] - anglesStraight[0])) -
user.getMeasurementsLittle().get(user.getMeasurementsLittle().size() - 2).getTamValue()) > 20) {
    warningActivated = true;
    builderWarningDialog.setMessage("Något av dina
värden avviker mycket från din förra mätning.\nVill du spara ändå?");
}
}
break;

}

if ((anglesStraight[0] == 0 && anglesStraight[1]==0) ||
(anglesBent[0] == 0 && anglesBent[1]==0)) {
    warningActivated= true;
    builderWarningDialog.setMessage("Du har inte flyttat på vissa
av prickarna, är du säker på att du vill spara mätningen?");
}

if (anglesBent[0]>130 ||anglesBent[1]>110 || anglesBent[2]>140
||((anglesBent[1] - anglesStraight[1]) + (anglesBent[0] -
anglesStraight[0])>230)){
    warningActivated = true;
    builderWarningDialog.setMessage("Vinklarna översiger
normalvärdet, vill du spara ändå?");//TODO dissutera med handledare om vad
det ska stå
}

builderWarningDialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        checkNextFinger();
    }
});
builderWarningDialog.setNegativeButton("Spara ej data", new
DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int id) {
        removeLast();
        backToMain();
    }
});

if (warningActivated) {
    builderWarningDialog.create();

    builderWarningDialog.setOnDismissListener(new
DialogInterface.OnDismissListener() {
        @Override
        public void onDismiss(DialogInterface dialog) {
            if(anglesStraight != null || anglesBent != null){
                giveWarningsIfFault();
            }
        }
    });
}

```

```

        builderWarningDialog.show();
    } else {
        checkNextFinger();
    }
}

private void addToFinger(String dateString) {

    switch (user.getFingersAffected().get(currentFingerMeasure)) {

        case "PEKFINGER":
            user.addMeasurementIndex(new MeasurementInfo(dateString,
(int) anglesBent[1], (int) anglesStraight[1], (int) anglesBent[0], (int)
anglesStraight[0], (int) anglesBent[2], (int) anglesStraight[2]));
            break;

        case "LÅNGFINGER":
            user.addMeasurementMiddle(new MeasurementInfo(dateString,
(int) anglesBent[1], (int) anglesStraight[1], (int) anglesBent[0], (int)
anglesStraight[0], (int) anglesBent[2], (int) anglesStraight[2]));
            break;

        case "RINGFINGER":
            user.addMeasurementRing(new MeasurementInfo(dateString,
(int) anglesBent[1], (int) anglesStraight[1], (int) anglesBent[0], (int)
anglesStraight[0], (int) anglesBent[2], (int) anglesStraight[2]));
            break;

        case "LILLFINGER":
            user.addMeasurementLittle(new MeasurementInfo(dateString,
(int) anglesBent[1], (int) anglesStraight[1], (int) anglesBent[0], (int)
anglesStraight[0], (int) anglesBent[2], (int) anglesStraight[2]));
            break;
    }
}

private void removeLast() {

    switch (user.getFingersAffected().get(currentFingerMeasure)) {

        case "PEKFINGER":

user.getMeasurementsIndex().remove(user.getMeasurementsIndex().get(user.get
MeasurementsIndex().size()-1));
            break;

        case "LÅNGFINGER":

user.getMeasurementsMiddle().remove(user.getMeasurementsMiddle().get(user.g
etMeasurementsMiddle().size()-1));
            break;

        case "RINGFINGER":

user.getMeasurementsRing().remove(user.getMeasurementsRing().get(user.getMe
asurementsRing().size()-1));
            break;

        case "LILLFINGER":
    }
}

```

```
user.getMeasurementsLittle().remove(user.getMeasurementsLittle().get(user.getMeasurementsLittle().size()-1));
    break;

}

}

private void checkNextFinger() {
    if (currentFingerMeasure<user.getFingersAffected().size()-1) {
        anglesStraight=null;
        anglesBent = null;
        currentFingerMeasure++;
        firstCameraStarted = false;
        secondCameraStarted = false;
        wantInfo = originalWantInfo;
        startMeasure();
    } else {
        backToMain();
    }
}

private void backToMain() {
    Intent mainIntent = new Intent(getApplicationContext(),
BaseActivity.class);
    startActivity(mainIntent);
    finish();
}

}
```

ViewUserActivity:

```
package com.example.digitalgoniometer;

import androidx.appcompat.app.ActionBar;
import android.content.Intent;
import android.content.res.Resources;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class ViewUserActivity extends BaseActivity {

    private TextView handInfo, fingerInfo, userNameInfo;
    private Button backButton, editButton;
    private ImageView fingerInfoImageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_user);

        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }

        checkSharedPreferences();

        backButton = findViewById(R.id.backButton);
        editButton = findViewById(R.id.editButton);
        handInfo = findViewById(R.id.handInformationTextView);
        fingerInfo = findViewById(R.id.fingerInfoTextView);
        userNameInfo = findViewById(R.id.userNameTextView);
        fingerInfoImageView = findViewById(R.id.fingerInfoImage);

        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent backIntent = new Intent(ViewUserActivity.this,
                    BaseActivity.class);
                startActivity(backIntent);
                finish();
            }
        });

        editButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent editIntent = new Intent(ViewUserActivity.this,
                    EditUserActivity.class);
                editIntent.putExtra("FROM_ACTIVITY", "VIEW_USER");
                startActivity(editIntent);
                finish();
            }
        });

        updateView();
    }
}
```

```

}

private void updateView() {
    userNameInfo.setText("Användarnamn: " + user.getUserName());

    if (user != null) {
        if (user.isRightHand()) {
            handInfo.setText("Skadad hand: Höger");
        } else {
            handInfo.setText("Skadad hand: Vänster");
        }
        if (user.getFingersAffected().size() > 0) {
            setImage();
        }
    } else {
        Toast toast = Toast.makeText(ViewUserActivity.this, "Hittar
ingen användare", Toast.LENGTH_SHORT);
        toast.show();
    }
}

private void setImage() {

    Resources res = getResources();

    if (user.isRightHand()) {

        if (!user.getFingersAffected().contains("LILLFINGER") &&
            !user.getFingersAffected().contains("RINGFINGER") &&
            !user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

            fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.h_no));
        }
        if (user.getFingersAffected().contains("LILLFINGER") &&
            !user.getFingersAffected().contains("RINGFINGER") &&
            !user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

            fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.h_lill));
        }
        if (user.getFingersAffected().contains("LILLFINGER") &&
            user.getFingersAffected().contains("RINGFINGER") &&
            !user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

            fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.h_lill_ring));
        }
        if (user.getFingersAffected().contains("LILLFINGER") &&
            user.getFingersAffected().contains("RINGFINGER") &&
            user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

            fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.h_lill_lang));
        }
        if (!user.getFingersAffected().contains("LILLFINGER") &&
            user.getFingersAffected().contains("RINGFINGER") &&
            user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

            fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.h_ring_lang));
        }
    }
}

```

```

        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_ring));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_ring_lang));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_ring_lang_pek));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lang));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        !user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lang_pek));
    }
    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lill_lang));
    }
    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lill_lang_pek));
    }
    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&

```

```

        !user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lill_pek));
}

if (user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    !user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_lill_ring_pek));
}

if (!user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    !user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_ring_pek));
}

if (user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.h_all));
}

} else {
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        !user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

        fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.v_no));
    }

    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        !user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

        fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.v_lill));
    }

    if (user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        !user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

        fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.v_lill_ring));
    }

    if (user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

        fingerInfoImageView.setImageResource(res.getDrawable(R.drawable.v_lill_ring_lang));
    }
}

```

```

        }
        if (!user.getFingersAffected().contains("LILLFINGER") &&
            user.getFingersAffected().contains("RINGFINGER") &&
            !user.getFingersAffected().contains("LÅNGFINGER") &&
            !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_ring));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_ring_lang));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_ring_lang_pek));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lang));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lang_pek));
    }
    if (!user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        !user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_pek));
    }
    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        !user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lill_lang));
    }
    if (user.getFingersAffected().contains("LILLFINGER") &&
        !user.getFingersAffected().contains("RINGFINGER") &&
        user.getFingersAffected().contains("LÅNGFINGER") &&
        user.getFingersAffected().contains("PEKFINGER")) {

fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lill_lang_pek));

```

```

        }
        if (user.getFingersAffected().contains("LILLFINGER") &&
            !user.getFingersAffected().contains("RINGFINGER") &&
            !user.getFingersAffected().contains("LÅNGFINGER") &&
            user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lill_pek));
}

if (user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    !user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_lill_ring_pek));
}

if (!user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    !user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_ring_pek));
}

if (user.getFingersAffected().contains("LILLFINGER") &&
    user.getFingersAffected().contains("RINGFINGER") &&
    user.getFingersAffected().contains("LÅNGFINGER") &&
    user.getFingersAffected().contains("PEKFINGER")) {

    fingerInfoImageView.setImageDrawable(res.getDrawable(R.drawable.v_all));
}
}
}
}

```

DateFormatter (i paket *model*):

```
package com.example.digitalgoniometer.model;
import java.text.FieldPosition;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateFormatter extends SimpleDateFormat {

    private String[] mLabels;

    public DateFormatter(String[] labels) {
        mLabels = labels;
    }

    @Override
    public StringBuffer format(Date date, StringBuffer buffer,
FieldPosition position) {

        int index = (int) date.getTime();

        String label = mLabels[index];

        buffer.append(label);

        return buffer;
    }
}
```

Dot (i paket model):

```
package com.example.digitalgoniometer.model;

public class Dot {

    private float x;
    private float y;
    private float radius;
    private boolean inMotion;

    public Dot(float x, float y, float radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
        this.setMotion(false);
    }

    public float getX() {
        return x;
    }

    public float getY() {
        return y;
    }

    public boolean areInMotion(){return inMotion;}

    public float getRadius() {
        return radius;
    }

    public void setX(float x){
        this.x = x;
    }
    public void setY(float y){
        this.y = y;
    }

    public void setMotion(boolean inMotion){this.inMotion = inMotion; }

    public void setRadius(int radius){this.radius = radius; }

    public boolean isInside(float x, float y) {
        return (getX() - x) * (getX() - x) + (getY() - y) * (getY() - y) <= radius * radius;
    }

}
```

HistoryInfo (i paket *model*):

```
package com.example.digitalgoniometer.model;

import java.util.ArrayList;

public class HistoryInfo {
    private ArrayList<String> fingersChecked, graphsChecked;

    public HistoryInfo() {
        fingersChecked = new ArrayList<>(4);
        graphsChecked = new ArrayList<>(7);
    }

    public ArrayList<String> getFingersChecked() {
        return fingersChecked;
    }

    public void setFingersChecked(ArrayList<String> fingersChecked) {
        this.fingersChecked = fingersChecked;
    }

    public ArrayList<String> getGraphsChecked() {
        return graphsChecked;
    }

    public void setGraphsChecked(ArrayList<String> graphsChecked) {
        this.graphsChecked = graphsChecked;
    }
}
```

MeasurementInfo (i paket *model*):

```
package com.example.digitalgoniometer.model;

public class MeasurementInfo {

    private String date;
    private int pipAngleBent, pipAngleStraight, dipAngleBent,
dipAngleStraight, mcpAngleBent, mcpAngleStraight, tamValue;

    public MeasurementInfo(String date, int pipAngleBent, int
pipAngleStraight, int dipAngleBent, int dipAngleStraight, int mcpBent, int
mcpStraight) {
        this.date = date;
        this.pipAngleBent = pipAngleBent;
        this.pipAngleStraight = pipAngleStraight;
        this.dipAngleBent = dipAngleBent;
        this.dipAngleStraight = dipAngleStraight;
        this.mcpAngleBent = mcpBent;
        this.mcpAngleStraight = mcpStraight;
        calculateTAMvalue();
    }

    public String getDate() {
        return date;
    }

    public int getPipAngleBent() {
        return pipAngleBent;
    }

    public int getPipAngleStraight() {
        return pipAngleStraight;
    }

    public int getDipAngleBent() {
        return dipAngleBent;
    }

    public int getDipAngleStraight() {
        return dipAngleStraight;
    }

    public int getMcpAngleBent() {
        return mcpAngleBent;
    }

    public int getMcpAngleStraight() {
        return mcpAngleStraight;
    }

    public int getTamValue() {
        return tamValue;
    }

    private void calculateTAMvalue() {
        tamValue = (pipAngleBent - pipAngleStraight) + (dipAngleBent -
dipAngleStraight);
    }
}
```

User (i paket model):

```
package com.example.digitalgoniometer.model;

import com.google.gson.annotations.SerializedName;
import java.util.ArrayList;

public class User {

    @SerializedName("hand")
    private boolean rightHand;
    @SerializedName("fingersAffectedArr")
    private ArrayList<String> fingersAffected;
    @SerializedName("measurementsArrPinky")
    private ArrayList<MeasurementInfo> measurementsLittle;
    @SerializedName("measurementsArrRing")
    private ArrayList<MeasurementInfo> measurementsRing;
    @SerializedName("measurementsArrMiddle")
    private ArrayList<MeasurementInfo> measurementsMiddle;
    @SerializedName("measurementsArrIndex")
    private ArrayList<MeasurementInfo> measurementsIndex;
    @SerializedName("userName")
    private String userName;

    public User() {
        fingersAffected = new ArrayList<>();
        measurementsLittle = new ArrayList<>();
        measurementsRing = new ArrayList<>();
        measurementsMiddle = new ArrayList<>();
        measurementsIndex = new ArrayList<>();
    }

    public ArrayList<String> getFingersAffected() {
        return fingersAffected;
    }
    public void setFingersAffected(ArrayList<String> fingersAffected) {
        this.fingersAffected = fingersAffected;
    }

    public ArrayList<MeasurementInfo> getMeasurementsLittle() {
        return measurementsLittle;
    }

    public ArrayList<MeasurementInfo> getMeasurementsRing() {
        return measurementsRing;
    }

    public ArrayList<MeasurementInfo> getMeasurementsMiddle() {
        return measurementsMiddle;
    }

    public ArrayList<MeasurementInfo> getMeasurementsIndex() {
        return measurementsIndex;
    }

    public boolean isRightHand() {
        return rightHand;
    }

    public void setRightHand(boolean rightHand) {
        this.rightHand = rightHand;
    }
}
```

```
}

public void addMeasurementLittle(MeasurementInfo measurementInfoToAdd) {
    measurementsLittle.add(measurementInfoToAdd);
}

public void addMeasurementRing(MeasurementInfo measurementInfoToAdd) {
    measurementsRing.add(measurementInfoToAdd);
}

public void addMeasurementMiddle(MeasurementInfo measurementInfoToAdd) {
    measurementsMiddle.add(measurementInfoToAdd);
}

public void addMeasurementIndex(MeasurementInfo measurementInfoToAdd) {
    measurementsIndex.add(measurementInfoToAdd);
}

public void changeLastMeasurementLittle(MeasurementInfo newMeasurement) {
    measurementsLittle.set(measurementsLittle.size()-1,newMeasurement);
}

public void changeLastMeasurementRing(MeasurementInfo newMeasurement) {
    measurementsRing.set(measurementsRing.size()-1,newMeasurement);
}

public void changeLastMeasurementMiddle(MeasurementInfo newMeasurement) {
    measurementsMiddle.set(measurementsMiddle.size()-1,newMeasurement);
}

public void changeLastMeasurementIndex(MeasurementInfo newMeasurement) {
    measurementsIndex.set(measurementsIndex.size()-1,newMeasurement);
}

public void setUserName(String userName) {
    this.userName = userName;
}

public String getUserName() {
    return userName;
}
}
```

CustomizedImageView (i paket *views*):

```
package com.example.digitalgoniometer.views;

import com.example.digitalgoniometer.MeasureActivity;
import com.example.digitalgoniometer.R;
import com.example.digitalgoniometer.model.Dot;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import java.util.ArrayList;

public class CustomizedImageView extends
    androidx.appcompat.widget.AppCompatImageView implements
    View.OnTouchListener {
    private final ArrayList<Dot> dots = new ArrayList<>();
    private Paint dotPaint;
    private Dot touchedDot;

    public CustomizedImageView(@NonNull Context context) {
        super(context);
        setup();
    }

    public CustomizedImageView(@NonNull Context context, @Nullable
        AttributeSet attrs) {
        super(context, attrs);
        setup();
    }

    public CustomizedImageView(@NonNull Context context, @Nullable
        AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        setup();
    }

    private void setup() {
        setOnTouchListener(this);
        dotPaint = new Paint();
        dotPaint.setColor(getResources().getColor(R.color.stockholm_blue));
        dotPaint.setAlpha(500);
    }

    public void setDotsOnScreen(boolean rightHand) {
        if (rightHand) {
            for (int i = 0; i < 5; i++) {
                dots.add(new Dot((i * -120) + 700, 200, 50));
            }
        } else {
            for (int i = 0; i < 5; i++) {
                dots.add(new Dot((i * 120) + 100, 200, 50));
            }
        }
        invalidate();
    }
}
```

```

    }

    private void drawConnectionLines(Canvas canvas) {
        canvas.drawLine(dots.get(0).getX(), dots.get(0).getY(),
dots.get(1).getX(), dots.get(1).getY(), dotPaint);
        canvas.drawLine(dots.get(1).getX(), dots.get(1).getY(),
dots.get(2).getX(), dots.get(2).getY(), dotPaint);
        canvas.drawLine(dots.get(2).getX(), dots.get(2).getY(),
dots.get(3).getX(), dots.get(3).getY(), dotPaint);
        canvas.drawLine(dots.get(3).getX(), dots.get(3).getY(),
dots.get(4).getX(), dots.get(4).getY(), dotPaint);
    }

    private void putTextOnDots(Canvas canvas, Paint textPaint){
        textPaint.setTextSize(33);

        Dot dot1 = dots.get(0);
        Dot dot2 = dots.get(1);
        Dot dot3 = dots.get(2);
        Dot dot4 = dots.get(3);
        Dot dot5 = dots.get(4);

        canvas.drawText("HAND", dot1.getX()-40, dot1.getY()+10, textPaint);
        canvas.drawText("MCP", dot2.getX()-35, dot2.getY()+10, textPaint);
        canvas.drawText("PIP", dot3.getX()-25, dot3.getY()+10, textPaint);
        canvas.drawText("DIP", dot4.getX()-25, dot4.getY()+10, textPaint);
        canvas.drawText("TOP", dot5.getX()-33, dot5.getY()+10, textPaint);
    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                dots.forEach(dot) -> {
                    if (dot.isInside(event.getX(), event.getY())) {
                        touchedDot = dot;
                        MeasureActivity.setZoomedBitmap(getBitmap ((int)
event.getX(),(int) event.getY()));
                    }
                );
                break;
            case MotionEvent.ACTION_MOVE:
                if (touchedDot != null) {
                    touchedDot.setX(event.getX());
                    touchedDot.setY(event.getY());
                    touchedDot.setRadius(100);
                    touchedDot.setMotion(true);

                    MeasureActivity.setZoomedBitmap(getBitmap ((int)
event.getX(),(int) event.getY()));
                    invalidate();
                }
                break;
            case MotionEvent.ACTION_UP:
                if (touchedDot != null) {
                    touchedDot.setRadius(50);
                    touchedDot.setMotion(false);
                    invalidate();
                    touchedDot = null;
                    MeasureActivity.resetZoomedBitmap();
                }
        }
    }
}

```

```

        break;
    case MotionEvent.ACTION_CANCEL:
        break;
    default:
        break;
    }
    return true;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);

    dots.forEach((dot) -> {
        if (!dot.areInMotion()) {
            dotPaint.setStyle(Paint.Style.FILL);
            canvas.drawCircle(dot.getX(), dot.getY(), dot.getRadius(),
dotPaint);
            drawConnectionLines(canvas);
            putTextOnDots(canvas, new
Paint(getApplicationContext().getColor(R.color.black)));
        } else {
            dotPaint.setStyle(Paint.Style.STROKE);
            dotPaint.setStrokeWidth(7);
            canvas.drawCircle(dot.getX(), dot.getY(), dot.getRadius(),
dotPaint);
            canvas.drawLine(dot.getX()-50, dot.getY()-50,
dot.getX()+50, dot.getY()+50, dotPaint);
            canvas.drawLine(dot.getX()+50, dot.getY()-50, dot.getX()-
50, dot.getY()+50, dotPaint);
        }
    });
}

public ArrayList<Dot> getCoordinates () {
    return dots;
}

public void resetCoordinates(){
    dots.clear();
}

private Bitmap getBitmap(int x, int y) {
    Bitmap originalBitmap = ((BitmapDrawable)
this.getDrawable()).getBitmap();
    Bitmap croppedBitmap;

    int bitmapWidth = originalBitmap.getWidth();
    int bitmapHeight = originalBitmap.getHeight();
    int imageViewWidth = this.getWidth();
    int imageViewHeight = this.getHeight();
    float scaleFactorx = (float) imageViewWidth / bitmapWidth;
    float scaleFactory = (float) imageViewHeight / bitmapHeight;

    Bitmap scaledBitmap = Bitmap.createScaledBitmap(originalBitmap,
(int) (bitmapWidth * scaleFactor), (int) (bitmapHeight * scaleFactory),
true);

    int imageWidthx = 100;
    int imageWidthy = 100;
}

```

```

        if (x <= 0 || y <= 0 || x >= this.getWidth() || y >=
this.getHeight()) {
            croppedBitmap = null;
        } else
            if ((x > imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) < this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
            croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
        } else
            if ((x <= imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) < this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
                imageWidthx = x;
                croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
            } else
                if ((x > imageWidthx) && (y <= imageWidthy) && ((x +
imageWidthx) < this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
                    imageWidthy=y;
                    croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
                } else
                    if ((x > imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) >= this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
                        imageWidthx=this.getWidth()-x;
                        croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
                    } else
                        if ((x > imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) < this.getWidth() && ((y + imageWidthy) >= this.getHeight())))
{
                            imageWidthy=this.getHeight()-y;
                            croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
                        } else
                            if ((x <= imageWidthx) && (y <= imageWidthy) && ((x +
imageWidthx) < this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
                                imageWidthx = x;
                                imageWidthy= y;
                                croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
                            } else
                                if ((x > imageWidthx) && (y <= imageWidthy) && ((x +
imageWidthx) >= this.getWidth() && ((y + imageWidthy) < this.getHeight())))
{
                                    imageWidthx=this.getWidth()-x;
                                    imageWidthy= y;
                                    croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
                                } else
                                    if ((x > imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) >= this.getWidth() && ((y + imageWidthy) >=
this.getHeight()))) {
                                        imageWidthx=this.getWidth()-x;
                                        imageWidthy=this.getHeight()-y;
                                        croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -

```

```
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
    } else
        if ((x <= imageWidthx) && (y > imageWidthy) && ((x +
imageWidthx) < this.getWidth()) && ((y + imageWidthy) >= this.getHeight()))
{
    imageWidthx=x;
    imageWidthy=this.getHeight()-y;
    croppedBitmap = Bitmap.createBitmap(scaledBitmap, x -
imageWidthx, y - imageWidthy, imageWidthx * 2, imageWidthy * 2);
} else {
    croppedBitmap=null;
}

int zoomFactor = 1;
Bitmap zoomedBitmap = null;
if (croppedBitmap!=null) {
    zoomedBitmap = Bitmap.createScaledBitmap(croppedBitmap,
croppedBitmap.getWidth() * zoomFactor, croppedBitmap.getHeight() *
zoomFactor, false);
}
return zoomedBitmap;
}
}
```

ZoomImageView (i paket *views*):

```
package com.example.digitalgoniometer.views;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.util.AttributeSet;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import com.example.digitalgoniometer.R;

public class ZoomImageView extends
    androidx.appcompat.widget.AppCompatImageView {
    private Paint crossPaint;
    public ZoomImageView(@NonNull Context context) {
        super(context);
        setup();
    }
    public ZoomImageView(@NonNull Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        setup();
    }
    public ZoomImageView(@NonNull Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        setup();
    }
    private void setup() {
        crossPaint = new Paint();
        crossPaint.setColor(getResources().getColor(R.color.stockholm_blue));
        crossPaint.setAlpha(500);
        invalidate();
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);

        crossPaint.setStyle(Paint.Style.STROKE);
        crossPaint.setStrokeWidth(7);
        canvas.drawLine((this.getWidth()/2)-50, (this.getHeight()/2)-50,
            (this.getWidth()/2)+50, (this.getHeight()/2)+50, crossPaint);
        canvas.drawLine((this.getWidth()/2)+50, (this.getHeight()/2)-50,
            (this.getWidth()/2)-50, (this.getHeight()/2)+50, crossPaint);
    }
}
```

activity_base.xml (i res, paket layout):

```
<?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"
    android:background="@color/stockholm_blue_light"
    tools:context=".MeasureActivity">

    <Button
        android:id="@+id/buttonBackMain"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="5dp"
        android:background="@drawable/baseline_arrow_back_24"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"/>

    <Button
        android:id="@+id/UserButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="5dp"
        android:background="@drawable/baseline_person_24"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:shadowColor="@color/design_default_color_primary_variant"/>

    <Button
        android:id="@+id/infoButtonMain"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="5dp"
        android:background="@drawable/baseline_info_24"
        app:layout_constraintTop_toBottomOf="@+id/UserButton"
        app:layout_constraintEnd_toEndOf="parent"
        android:shadowColor="@color/design_default_color_primary_variant"/>

    <ImageView
        android:id="@+id/startImage"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintTop_toBottomOf="@+id/guideline5"
        app:layout_constraintEnd_toEndOf="@+id/guideline18"
        app:layout_constraintStart_toStartOf="@+id/guideline17"
        app:layout_constraintBottom_toBottomOf="@+id/guideline16"
        android:background="@drawable/start_image"
        android:scaleType="fitCenter"
        android:adjustViewBounds="true"
        android:scaleY="0.8"
        android:scaleX="0.8"/>

    <Button
        android:id="@+id/historyButton"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Visa Historik"
        android:textSize="20dp"
```

```
        android:paddingBottom="20dp"
        android:paddingTop="20dp"
        app:layout_constraintTop_toBottomOf="@+id/guideline16"
        app:layout_constraintEnd_toEndOf="@+id/guideline15"
        app:layout_constraintStart_toStartOf="@+id/guideline14"

    android:shadowColor="@color/design_default_color_primary_variant"/>
    <Button
        android:id="@+id/measureButton"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Starta ny mätning"
        android:textSize="20dp"
        android:paddingBottom="20dp"
        android:paddingTop="20dp"
        app:layout_constraintTop_toBottomOf="@+id/historyButton"
        app:layout_constraintEnd_toEndOf="@+id/guideline15"
        app:layout_constraintStart_toStartOf="@+id/guideline14"

    android:shadowColor="@color/design_default_color_primary_variant"/>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.08" />
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.60" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.18" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.82" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.08" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
```

```
    app:layout_constraintGuide_percent="0.92" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_edit_history.xml (i res, paket layout):

```
<?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"
    android:background="@color/stockholm_blue_light"
    tools:context=".EditHistoryActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Välj vilka grafer du vill visa:"
        android:textStyle="bold"
        android:textSize="20dp"
        android:textColor="@color/ap_black"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/guideline6"
        android:id="@+id/editHeading"
    />

    <TextView
        android:id="@+id/fingerChoice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Finger:"
        android:textSize="20dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:textColor="@color/ap_black"
        app:layout_constraintTop_toBottomOf="@+id/editHeading"
        app:layout_constraintStart_toStartOf="@+id/guideline12"
    />

    <LinearLayout
        android:id="@+id/checkboxesFingers"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:gravity="center_horizontal"
        app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintTop_toBottomOf="@+id/checkboxesFingers"
            app:layout_constraintStart_toStartOf="parent"
            android:id="@+id/linearLayoutH1"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:layout_gravity="center"
```

```
>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistIndex"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pekfinger"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH1"
    android:id="@+id/linearLayoutH2"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
    >
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistMiddle"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Långfinger"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH2"
    android:id="@+id/linearLayoutH3"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
    >
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistRing"
/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ringfinger"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH3"
    android:id="@+id/linearLayoutH4"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistLittle"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Lillfinger"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
</LinearLayout>
</LinearLayout>

<TextView
    android:id="@+id/graphChoice"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Grafer:"
    android:textSize="20dp"
    android:paddingTop="20dp"
    android:paddingBottom="20dp"
    android:textColor="@color/ap_black"
    app:layout_constraintTop_toBottomOf="@+id/checkboxexFingers"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toBottomOf="@+id/graphChoice"
    android:orientation="vertical">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
```

```
    app:layout_constraintTop_toBottomOf="@+id/graphChoice"
    app:layout_constraintStart_toStartOf="@id/guideline12"
    android:id="@+id/linearLayoutH5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TAM: "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistTAM"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH5"
    app:layout_constraintStart_toStartOf="@id/guideline12"
    android:id="@+id/linearLayoutH6">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MCP: "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistMCPbent"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Böjd"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistMCPstraight"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" "
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />

```

```
        android:layout_height="wrap_content"
        android:text="Sträckt"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH6"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    android:id="@+id/linearLayoutH7">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DIP: "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistDIPbent"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Böjd"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistDIPstraight"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sträckt"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH7"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    android:id="@+id/linearLayoutH8">
```

```
        android:id="@+id/linearLayoutH8">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="PIP: "
            android:textStyle="bold"
            android:textSize="15dp"
            android:textColor="@color/ap_black"
            android:layout_gravity="center_vertical"
        />
        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="30dp"
            android:layout_gravity="center_vertical"
            android:id="@+id/checkBoxHistPIPbent"
        />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Böjd"
            android:textSize="15dp"
            android:textColor="@color/ap_black"
            android:layout_gravity="center_vertical"
        />
        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="30dp"
            android:layout_gravity="center_vertical"
            android:id="@+id/checkBoxHistPIPstraight"
        />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sträckt"
            android:textSize="15dp"
            android:textColor="@color/ap_black"
            android:layout_gravity="center_vertical"
        />
    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:id="@+id/buttonLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="bottom"
    app:layout_constraintBottom_toBottomOf="@+id/guideline7">

    <Button
        android:id="@+id/editHistCancelButton"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Avbryt"
        android:layout_marginEnd="15dp"
```

```
        android:layout_marginStart="20dp"/> >

    <Button
        android:id="@+id/editHistSaveButton"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Spara"
        android:layout_marginStart="15dp"
        android:layout_marginEnd="20dp"/> >

</LinearLayout>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="150dp" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.98" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.08" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.92" />

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

activity_edit_history.xml (land) (i res, paket layout):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/stockholm_blue_light"
        xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Välj vilka grafer du vill visa:"
        android:textStyle="bold"
        android:textSize="20dp"
        android:textColor="@color/ap_black"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/guideline6"
        android:id="@+id/editHeading"
    />

    <TextView
        android:id="@+id/fingerChoice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Finger:"
        android:textSize="20dp"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:textColor="@color/ap_black"
        app:layout_constraintTop_toBottomOf="@+id/editHeading"
        app:layout_constraintStart_toStartOf="@+id/guideline12"
    />

    <LinearLayout
        android:id="@+id/checkboxexFingers"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    >

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/checkboxexFingers"
            android:id="@+id/linearLayoutH1"
            android:paddingLeft="11dp"
            android:paddingRight="10dp"
            android:layout_gravity="center"
        >
            <CheckBox
                android:layout_width="wrap_content"
                android:layout_height="30dp"
                android:layout_gravity="center_vertical"
            >
        
```

```
        android:id="@+id/checkBoxHistIndex"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pekfinger"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH1"
    android:id="@+id/linearLayoutH2"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistMiddle"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Långfinger"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH2"
    android:id="@+id/linearLayoutH3"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistRing"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ringfinger"
        android:textSize="15dp"
    />
```

```
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@+id/fingerChoice"
    app:layout_constraintStart_toEndOf="@+id/linearLayoutH3"
    android:id="@+id/linearLayoutH4"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:layout_gravity="center"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistLittle"
        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lillfinger"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
        />
</LinearLayout>
</LinearLayout>

<TextView
    android:id="@+id/graphChoice"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Grafer:"
    android:textSize="20dp"
    android:paddingStart="90dp"
    android:paddingBottom="10dp"
    android:textColor="@color/ap_black"
    app:layout_constraintTop_toBottomOf="@+id/guideline6"
    app:layout_constraintStart_toEndOf="@+id/fingerChoice"
    />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="@+id/graphChoice"
    app:layout_constraintTop_toBottomOf="@+id/graphChoice"
    android:paddingStart="100dp"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintTop_toBottomOf="@+id/graphChoice"
        app:layout_constraintStart_toStartOf="@+id/guideline12"
        android:id="@+id/linearLayoutH5">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TAM: "
    android:textStyle="bold"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistTAM"
/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH5"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    android:id="@+id/linearLayoutH6">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="MCP: "
    android:textStyle="bold"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistMCPbent"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Böjd"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistMCPstraight"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sträckt"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
```

```
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH6"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    android:id="@+id/linearLayoutH7">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DIP: "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistDIPbent"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Böjd"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistDIPstraight"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sträckt"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@+id/linearLayoutH7"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    android:id="@+id/linearLayoutH8">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBoxHistDIPbent"
    />
```

```
        android:layout_height="wrap_content"
        android:text="PIP:    "
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistPIPBent"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Böjd"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center_vertical"
    android:id="@+id/checkBoxHistPIPstraight"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sträckt"
    android:textSize="15dp"
    android:textColor="@color/ap_black"
    android:layout_gravity="center_vertical"
/>
</LinearLayout>

</LinearLayout>

<LinearLayout
    android:id="@+id/buttonLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="bottom"
    app:layout_constraintBottom_toBottomOf="@+id/guideline7">

    <Button
        android:id="@+id/editHistCancelButton"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Avbryt"
        android:layout_marginEnd="15dp"
        android:layout_marginStart="20dp"/>

    <Button
        android:id="@+id/editHistSaveButton"
```

```
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Spara"
        android:layout_marginStart="15dp"
        android:layout_marginEnd="20dp"/>

    </LinearLayout>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="50dp" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.98" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.08" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.10" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.92" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_edit_user.xml (i res, paket layout):

```
<?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"
        android:background="@color/stockholm_blue_light"
        tools:context=".EditUserActivity">

    <Button
        android:id="@+id/removeUserButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_delete_forever_24"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Redigera användare"
        android:textSize="30dp"
        android:textColor="@color/ap_black"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/guideline6"
        android:id="@+id/userHeading"
    />
    <LinearLayout
        android:id="@+id/userNameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintTop_toBottomOf="@+id/guideline6"
        app:layout_constraintBottom_toTopOf="@+id/handInformationTextView"
        app:layout_constraintStart_toStartOf="@+id/guideline12">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Användarnamn:"
            android:textSize="20dp"
            android:textColor="@color/ap_black"
        />

        <EditText
            android:id="@+id/userNameInputView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Skriv här..."
            android:textColor="@color/ap_gray"
            android:textSize="15dp"/>
    
```

```
</LinearLayout>

<TextView
    android:id="@+id/handInformationTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Skadad hand:"
    android:textSize="20dp"
    android:textColor="@color/ap_black"
    app:layout_constraintTop_toBottomOf="@+id/userNameTextView"
    app:layout_constraintBottom_toTopOf="@+id/switchLinearLayout"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    />

<LinearLayout
    android:id="@+id/switchLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    app:layout_constraintTop_toBottomOf="@+id/handInformationTextView"
    app:layout_constraintBottom_toTopOf="@+id/fingerInfoTextView"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Vänster"
        android:textSize="20dp"
        android:layout_marginEnd="10dp"/>

    <Switch
        android:id="@+id/leftRightSwitch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:thumbTint="@color/stockholm_blue"
        android:trackTint="@color/stockholm_blue"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Höger"
        android:textSize="20dp"
        android:layout_marginStart="10dp"/>

</LinearLayout>

<TextView
    android:id="@+id/fingerInfoTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20dp"
    android:text="Skadade fingrar: "
    android:layout_weight="0.5"
    android:textColor="@color/ap_black"
    app:layout_constraintTop_toBottomOf="@+id/switchLinearLayout"
    app:layout_constraintBottom_toTopOf="@+id/handImageView"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
```

```
    />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/fingerInfoTextView"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    android:id="@+id/linearLayout1"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBox1"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pekfinger"
        android:textSize="15dp"
        android:paddingLeft="20dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
        />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout1"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout3"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    android:id="@+id/linearLayout2"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBox2"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Långfinger"
        android:textSize="15dp"
        android:paddingLeft="20dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
        />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout2"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout4"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    android:id="@+id/linearLayout3"
    >
    <CheckBox
```

```
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBox3"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ringfinger"
        android:textSize="15dp"
        android:paddingLeft="20dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout3"
    app:layout_constraintBottom_toTopOf="@+id/buttonLinearLayout"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    android:id="@+id/linearLayout4"
    >
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:id="@+id/checkBox4"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lillfinger"
        android:textSize="15dp"
        android:paddingLeft="20dp"
        android:textColor="@color/ap_black"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>

<LinearLayout
    android:id="@+id/buttonLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="bottom"
    app:layout_constraintBottom_toBottomOf="@+id/guideline7">

    <Button
        android:id="@+id/CancelButton"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Avbryt"
        android:layout_marginEnd="15dp"
        android:layout_marginStart="20dp"/>

    <Button
        android:id="@+id/saveButton"
```

```
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="Spara"
        android:layout_marginStart="15dp"
        android:layout_marginEnd="20dp"/>

    </LinearLayout>

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.40" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.98" />
    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.2" />
    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.08" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.92" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_history.xml (i res, paket layout):

```
<?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"
        app:layout_constraintStart_toStartOf="parent"
        tools:context=".HistoryActivity"
        android:background="@color/stockholm_blue_light"
        android:id="@+id/constraintLayout">

    <Button
        android:id="@+id/editHistoryButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_settings_24"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/historyBackButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_arrow_back_24"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <LinearLayout
        android:id="@+id/switchLinearLayoutHistory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:gravity="center_horizontal"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            android:text="GRAF"
            android:textSize="20dp"
            android:layout_marginEnd="10dp"/>

    <Switch
        android:layout_width="55dp"
        android:layout_height="40dp"
        android:id="@+id/leftRightSwitchHistory"
        android:checked="false"
        android:thumbTint="@color/stockholm_blue"
        android:trackTint="@color/stockholm_blue"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
    />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TABELL"
    android:textSize="20dp"
    android:layout_marginStart="10dp"/>

</LinearLayout>

<RelativeLayout
    android:id="@+id/relativeLayoutHistory"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/switchLinearLayoutHistory"
    app:layout_constraintBottom_toTopOf="@id/guideline13"
    app:layout_constraintEnd_toEndOf="parent"
    >

    <com.androidplot.xy.XYPlot
        android:id="@+id/graphViewPlot"
        style="@style/APDefacto.Light"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleX="0.9"
        android:scaleY="0.9"
        android:background="@android:color/transparent"
        android:backgroundTint="@null"
        />

    <ScrollView
        android:id="@+id/scrollViewTable"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TableLayout
            android:id="@+id/tableLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="gone"
            >

        </TableLayout>

    </ScrollView>
</RelativeLayout>

<LinearLayout
    android:id="@+id/colorLinearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/guideline13"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="TAM"
        android:textSize="10dp"
        android:textColor="@color/black"/>
```

```
<ImageView
    android:id="@+id/tamColorView"
    android:layout_margin="5dp"
    android:layout_width="13dp"
    android:layout_height="13dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="DIP"
    android:textSize="10dp"
    android:textColor="@color/black"/>

<ImageView
    android:id="@+id/dipColorView"
    android:layout_margin="5dp"
    android:layout_width="13dp"
    android:layout_height="13dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="PIP"
    android:textSize="10dp"
    android:textColor="@color/black"/>

<ImageView
    android:id="@+id/pipColorView"
    android:layout_width="13dp"
    android:layout_height="13dp"
    android:layout_margin="5dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="MCP"
    android:textSize="10dp"
    android:textColor="@color/black"/>

<ImageView
    android:id="@+id/mcpColorView"
    android:layout_margin="5dp"
    android:layout_width="13dp"
    android:layout_height="13dp"/>

</LinearLayout>

<androidx.appcompat.widget.LinearLayoutCompat
    android:id="@+id/colorLinearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/colorLinearLayout"
    app:layout_constraintBottom_toBottomOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_margin="5dp"
        android:text="BÖJT"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/bentLineView"
        android:layout_margin="5dp"
        android:layout_width="15dp"
        android:layout_height="15dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="STRÄCKT"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/straightLineView"
        android:layout_margin="5dp"
        android:layout_width="15dp"
        android:layout_height="15dp"/>

</androidx.appcompat.widget.LinearLayoutCompat>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.85" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.04" />

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

activity_history.xml (land) (i res, paket layout):

```
<?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"
        app:layout_constraintStart_toStartOf="parent"
        tools:context=".HistoryActivity"
        android:background="@color/stockholm_blue_light"
        android:id="@+id/constraintLayout">

    <Button
        android:id="@+id/editHistoryButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_settings_24"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/historyBackButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_arrow_back_24"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textGraph"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        app:layout_constraintStart_toEndOf="@+id/leftRightSwitchHistory"
            app:layout_constraintTop_toTopOf="parent"
            android:text="GRAF"
            android:textSize="20dp"
            android:layout_marginEnd="10dp"/>

    <Switch
        android:layout_width="55dp"
        android:layout_height="40dp"
        android:id="@+id/leftRightSwitchHistory"
        android:checked="true"
        android:thumbTint="@color/stockholm_blue"
        android:trackTint="@color/stockholm_blue"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"

    />

    <TextView
        android:id="@+id/textTable"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="TABELL"
        android:textSize="20dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/leftRightSwitchHistory"
        android:layout_marginStart="10dp"/>

    <RelativeLayout
        android:id="@+id/relativeLayoutHistory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

    app:layout_constraintTop_toBottomOf="@+id/leftRightSwitchHistory"
        app:layout_constraintBottom_toTopOf="@+id/guideline13"
        app:layout_constraintEnd_toEndOf="parent"
    >

        <com.androidplot.xy.XYPlot
            android:id="@+id/graphViewPlot"
            style="@style/APDefacto.Light"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleX="0.9"
            android:scaleY="0.9"
            android:background="@android:color/transparent"
            android:backgroundTint="@null"
        />

        <ScrollView
            android:id="@+id/scrollViewTable"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TableLayout
                android:id="@+id/tableLayout"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:visibility="gone"
            >

            </TableLayout>

        </ScrollView>
    </RelativeLayout>

    <LinearLayout
        android:id="@+id/colorLinearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/guideline13"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="5dp"
            android:layout_marginBottom="5dp"
            android:layout_marginStart="5dp"
            android:paddingStart="10dp"
            android:text="TAM"
            android:textSize="10dp"
```

```
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/tamColorView"
        android:layout_margin="5dp"
        android:layout_width="13dp"
        android:layout_height="13dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="5dp"
        android:paddingStart="15dp"
        android:text="DIP"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/dipColorView"
        android:layout_margin="5dp"
        android:layout_marginEnd="5dp"
        android:layout_width="13dp"
        android:layout_height="13dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="5dp"
        android:paddingStart="15dp"
        android:text="PIP"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/pipColorView"
        android:layout_width="13dp"
        android:layout_height="13dp"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="5dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="5dp"
        android:paddingStart="15dp"
        android:text="MCP"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/mcpColorView"
        android:layout_margin="5dp"
        android:layout_marginEnd="5dp"
```

```
        android:layout_width="13dp"
        android:layout_height="13dp"/>

    </LinearLayout>

<androidx.appcompat.widget.LinearLayoutCompat
    android:id="@+id/colorLinearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/colorLinearLayout"
    app:layout_constraintBottom_toBottomOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:paddingStart="10dp"
        android:text="BÖJT"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/bentLineView"
        android:layout_margin="5dp"
        android:layout_width="15dp"
        android:layout_height="15dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:paddingStart="15dp"
        android:text="STRÄCKT"
        android:textSize="10dp"
        android:textColor="@color/black"/>

    <ImageView
        android:id="@+id/straightLineView"
        android:layout_margin="5dp"
        android:layout_width="15dp"
        android:layout_height="15dp"/>

</androidx.appcompat.widget.LinearLayoutCompat>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.85" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.1" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_measure.xml (i res, paket layout):

```
<?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"
        android:background="@color/stockholm_blue_light"
        tools:context=".MainActivity">

    <com.example.digitalgoniometer.views.CustomizedImageView
        android:id="@+id/customizedImageView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/guideline4"
        app:layout_constraintStart_toStartOf="@+id/guideline"
        app:layout_constraintEnd_toEndOf="@+id/guideline2"
        app:layout_constraintTop_toBottomOf="@+id/infoTextMeasureView" />

    <ImageView
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:id="@+id/infoImageView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

    <com.example.digitalgoniometer.views.ZoomImageView
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:id="@+id/zoomImageView"
        android:background="@color/ap_white"
        android:paddingRight="5dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ZOOM"
        android:textColor="@color/ap_gray"
        android:textSize="14dp"
        app:layout_constraintStart_toStartOf="@+id/zoomImageView"
        app:layout_constraintEnd_toEndOf="@+id/zoomImageView"
        app:layout_constraintTop_toTopOf="@+id/zoomImageView"/>

    <TextView
        android:id="@+id/infoTextMeasureView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Du mäter nu ditt: "
        android:textSize="16dp"
        app:layout_constraintTop_toBottomOf="@+id/infoImageView"
        app:layout_constraintStart_toStartOf="@+id/guideline"
        app:layout_constraintEnd_toStartOf="@+id/guideline2"/>

    <Button
        android:id="@+id/infoButton"
        android:layout_width="30dp"
        android:layout_height="30dp"
```

```
        android:background="@drawable/baseline_info_24"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/infoTextMeasureView"/>

<LinearLayout
    android:id="@+id/LinearButtonContainer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    app:layout_constraintEnd_toEndOf="@+id/guideline2"
    app:layout_constraintStart_toStartOf="@+id/guideline"
    app:layout_constraintBottom_toBottomOf="parent">

    <Button
        android:id="@+id/cancelMeasureButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Avbryt"
        android:textSize="10dp"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="5dp"
    />

    <Button
        android:id="@+id/takeNewPictureButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ta ny bild"
        android:textSize="10dp"
        android:layout_marginRight="5dp"
        android:layout_marginLeft="5dp"
        android:layout_marginBottom="5dp"
    />

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Spara"
        android:textSize="20dp"
        android:layout_marginLeft="5dp"
        android:layout_marginBottom="5dp"
    />

</LinearLayout>

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.1" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.9" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.08" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.92" />

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

activity_view_user.xml (i res, paket layout):

```
<?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"
        android:background="@color/stockholm_blue_light"
        tools:context=".ViewUserActivity">

    <TextView
        android:id="@+id/userHeading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Användare"
        android:textColor="@color/ap_black"
        android:textSize="30dp"
        app:layout_constraintBottom_toTopOf="@+id/guideline8"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/userNameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Användarnamn:"
        android:textColor="@color/ap_black"
        android:textSize="20dp"

        app:layout_constraintStart_toStartOf="@+id/guideline10"
        app:layout_constraintTop_toBottomOf="@+id/guideline8" />

    <TextView
        android:id="@+id/handInformationTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Skadad hand:"
        android:textColor="@color/ap_black"
        android:textSize="20dp"
        android:layout_marginTop="20dp"
        app:layout_constraintStart_toStartOf="@+id/guideline10"
        app:layout_constraintTop_toBottomOf="@+id/userNameTextView"

        />

    <TextView
        android:id="@+id/fingerInfoTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Skadade fingrar: "
        android:textColor="@color/ap_black"
        android:textSize="20dp"
        android:layout_marginTop="20dp"
        app:layout_constraintStart_toStartOf="@+id/guideline10"
```

```
    app:layout_constraintTop_toBottomOf="@+id/handInformationTextView" />

    <ImageView
        android:id="@+id/fingerInfoImage"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintStart_toStartOf="@+id/guideline10"
        app:layout_constraintTop_toBottomOf="@+id/fingerInfoTextView"
        app:layout_constraintBottom_toBottomOf="@+id/guideline20"
        app:layout_constraintEnd_toEndOf="@+id/guideline19"
        android:adjustViewBounds="true"
        android:scaleType="fitCenter" />

    <Button
        android:id="@+id/backButton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@drawable/baseline_arrow_back_24"

        android:backgroundTint="@color/design_default_color_primary_variant"
        android:layout_margin="5dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/editButton"
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:background="@drawable/baseline_edit_24"

        android:backgroundTint="@color/design_default_color_primary_variant"
        android:layout_margin="5dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.25" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.2" />
    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.08" />
    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline19"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.92" />
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.98" />

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

main_info_dialog.xml (i res, paket layout):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:padding="25dp">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fadeScrollbars="false"
        >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="En skada på handens böjsenor påverkar böjförmågan hos
det påverkade fingret/fingrarna.&#10;Böjsenorna löper från underarmens
muskler ända ut till fingertopparna och fäster i fingrarnas
leder.&#10;Lederna kallas metakarpofalangealleden (MCP), proximala
interfalangealleden (PIP) samt distala interfalangealleden (DIP)."
                android:textSize="14dp" />

            <ImageView
                android:layout_width="200dp"
                android:layout_height="280dp"
                android:background="@drawable/joint_info"
                android:layout_margin="10dp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Du kan se historik från dina tidigare mätningar om du
klickar på knappen "Visa historik".&#10;&#10;Du startar en ny mätning genom
knappen "Starta mätning".&#10;&#10;Du kan se och redigera din profil genom
att trycka på "gubben" i högra hörnet."
                android:textSize="14dp" />

        </LinearLayout>
    </ScrollView>

</LinearLayout>
```

measure_info_dialog.xml (i res, paket layout):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="20dp"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fadeScrollbars="false"
    >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:id="@+id/mainText"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14dp"
                android:text="Du kommer nu få ta två foton per skadat finger&#10;&#10;Tänk på att hålla kameran rakt framför "lillfinger-sidan/Pekfingersidan" så det blir 90° mellan kameran och handen för korrekt mätresultat.

                &#10;&#10;Om en mätpunkt gömmer sig bakom ett annat finger får du uppskatta vart denna punkt borde sitta."
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toTopOf="parent"/>

            <ImageView
                android:id="@+id/jointInfoImage"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/mainText"
                android:scaleType="centerInside"
                android:foregroundGravity="center"

            />
        </LinearLayout>

    </ScrollView>

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

measure_info_second_dialog.xml (i res, paket layout):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:paddingBottom="5dp"
        android:paddingTop="10dp"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <ScrollView
        android:id="@+id/scrollViewSecond"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fadeScrollbars="false"
        >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:id="@+id/SecondHeader"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14dp"
                android:textStyle="bold"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <TextView
                android:id="@+id/SecondMainText"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="13dp"
                android:text="Du kommer nu få ta en bild på din skadade hand med
                fingrarna så sträckta som möjligt.

 Ta bilden så fingrarna syns
                väl med kameran rakt framför handens sida.&#10;&#10; Du ska därefter
                placera ut punkter på dina ledar på det skadade fingret.&#10;&#10;Flytta
                punkterna genom att dra dem på skärmen till rätt led. &#10;&#10;När du är
                nöjd med placeringen klickar du på spara."
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/SecondHeader" />

            <VideoView
                android:id="@+id/secondInfoVideo"
                android:layout_width="300dp"
                android:layout_height="400dp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/SecondMainText"

                android:scaleType="centerInside"
                android:foregroundGravity="center"
```

```
    android:layout_margin="10dp"
  />

</LinearLayout>

</ScrollView>

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

measure_info_third_dialog.xml (i res, paket layout):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:paddingBottom="5dp"
        android:paddingTop="10dp"
        xmlns:app="http://schemas.android.com/apk/res-auto">

    <ScrollView
        android:id="@+id/scrollViewThird"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fadeScrollbars="false"
        >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:id="@+id/thirdHeader"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14dp"
                android:textStyle="bold"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <TextView
                android:id="@+id/thirdMainText"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="13dp"
                android:text="Du kommer nu få ta en bild på din skadade hand med
fingrarna så böjda som möjligt.&#10;&#10; Ta bilden så fingrarna syns väl
med kameran rakt framför handens sida.&#10;&#10; Du ska därefter placera ut
punkter på dina ledar på det skadade fingret.&#10;&#10;Flytta punkterna
genom att dra dem på skärmen till rätt led. &#10;&#10;När du är nöjd med
placeringen klickar du på spara."
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/thirdHeader" />

            <VideoView
                android:id="@+id/thirdInfoVideo"
                android:layout_width="300dp"
                android:layout_height="400dp"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/thirdMainText"
```

```
        android:scaleType="centerInside"
        android:foregroundGravity="center"
        android:layout_margin="10dp"
    />

</LinearLayout>

</ScrollView>

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

colors.xml (i res, paket *values*):

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="stockholm_blue">#007FC8</color>
    <color name="stockholm_blue_light">#DDE9F8</color>
    <color name="orange">#FF6100</color>

</resources>
```

themes.xml (i res, paket values):

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.DigitalGoniometer"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/stockholm_blue</item>
        <item name="colorPrimaryVariant">@color/stockholm_blue_light</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item
name="android:statusBarColor">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item name="android:popupBackground">@android:color/white</item>
    </style>
</resources>
```

themes.xml (night) (i res, paket values):

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.DigitalGoniometer"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/stockholm_blue</item>
        <item name="colorPrimaryVariant">@color/stockholm_blue_light</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_200</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item
name="android:statusBarColor">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item
name="android:alertDialogTheme">@style/AlertDialogTheme</item>
    </style>

    <style name="AlertDialogTheme"
parent="ThemeOverlay.AppCompat.Dialog.Alert">
        <item name="android:windowBackground">@color/white</item>
        <item name="android:textColor">@color/black</item>
    </style>

</resources>
```

file_provider_paths.xml (in res, paket xml):

```
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-files-path name="my_images" path="Pictures/" />
</paths>
```

build.gradle (Module:app) (i paket *Gradle Scripts*):

```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.digitalgoniometer'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.digitalgoniometer"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.code.gson:gson:2.8.8'
    implementation 'com.androidplot:androidplot-core:1.5.10'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
}
```

AndroidManifest.xml (i paket manifests):

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

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/icon_image_4"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.DigitalGoniometer"
        tools:targetApi="31">

        <activity
            android:name=".EditUserActivity"
            android:exported="false"
            android:screenOrientation="portrait"/>
        <activity
            android:name=".HistoryActivity"
            android:exported="false"
            android:screenOrientation="landscape"
            />
        <activity
            android:name=".EditHistoryActivity"
            android:exported="false"
            android:screenOrientation="landscape"
            />
        <activity
            android:name=".ViewUserActivity"
            android:exported="false"
            android:screenOrientation="portrait"/>
        <activity
            android:name=".MeasureActivity"
            android:exported="false"
            android:screenOrientation="portrait"/>
        <activity
            android:name=".BaseActivity"
            android:exported="true"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>

        <provider
            android:name="androidx.core.content.FileProvider"

        android:authorities="com.example.digitalgoniometer.file_provider_paths"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
```

```
        android:resource="@xml/file_provider_paths" />
    </provider>
</application>

</manifest>
```