Mobile and Wearable Computing

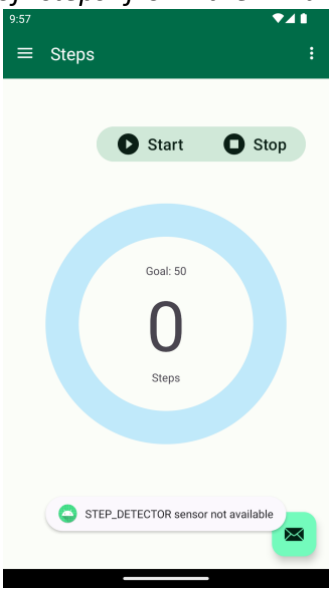
Assignment 03

Davide Grandesso

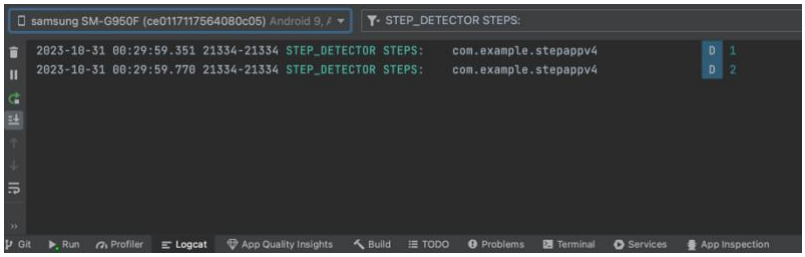
Exercise 1 – Step Counter

1. countSteps() & STEP_DETECTOR

Create the function countSteps() that counts the number of steps from the Android STEP_DETECTOR events and print them in the Logcat.



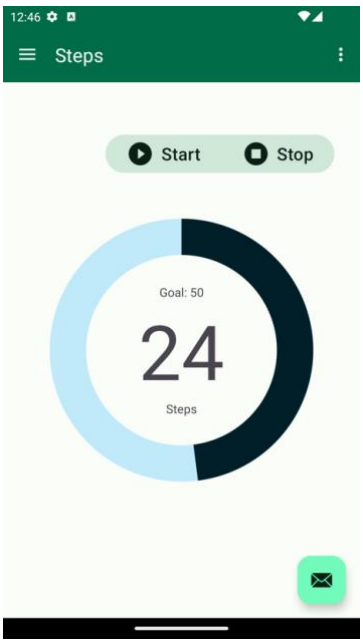

```
private void countSteps(float step)
{
    stepDetectorCounter += step;

    Log.d("STEP_DETECTOR STEPS: ",
String.valueOf(stepDetectorCounter));

}
```



STEP_DETECTOR sensor is not available on virtual device () but it's possible to use it on Samsung SM-G950F

2. Update the Circular Progress Bar

Update the Circular Progress Bar with the number of steps. You can use the number of steps calculated with the accelerometer or with the Android step detector.



Exercise 2 – Data storage and visualization – Daily Report

1. Add a "count" button Chart view

Create chart view in XML using the AnyChart library.

fragment_day.xml

```
<com.anychart.AnyChartView
    android:id="@+id/DayBarChart"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.31" />
```

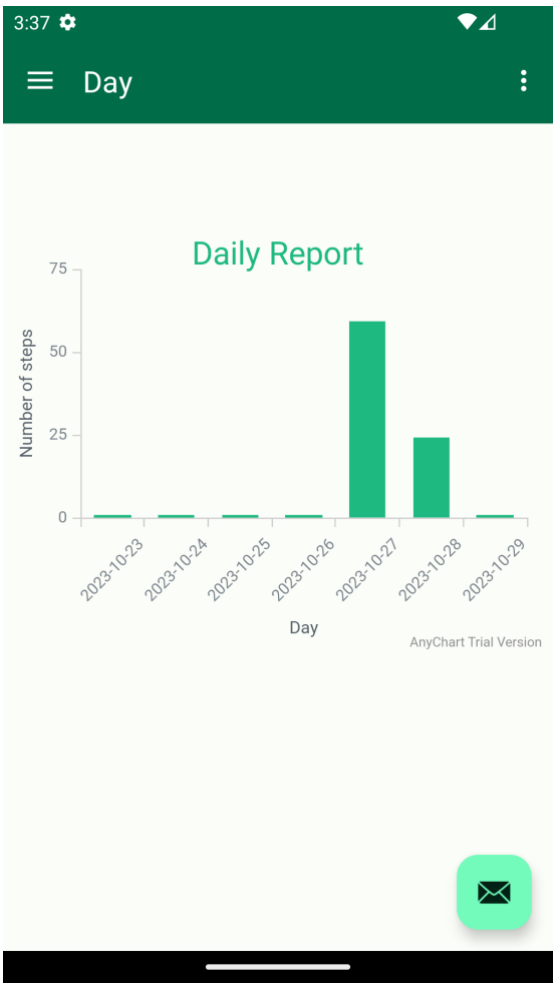
2. Data Retrive

Implement a method to retrieve the number of steps by date from the SQLite stepapp database.

StepAppOpenHelper.java

```
public static Map<String, Integer> loadStepsByDay(Context context, String
date){
    // 1. Define a map to store the hour and number of steps as key-value
pairs
    Map<String, Integer>  map = new TreeMap<>();

    // 2. Get the readable database
    StepAppOpenHelper databaseHelper = new StepAppOpenHelper(context);
    SQLiteDatabase database = databaseHelper.getReadableDatabase();


    //Manage Monday & Sunday
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-
dd");
    LocalDate data = LocalDate.parse(date, formatter);
    DayOfWeek dayOfWeek = data.getDayOfWeek();
    // Calculates the difference in days between the current day of the
week and Monday (DayOfWeek.MONDAY)
    int differenceDays = DayOfWeek.MONDAY.getValue() -
dayOfWeek.getValue();
    // Get the Monday of the week by subtracting the difference of days
from the date
    LocalDate monday = data.plusDays(differenceDays);
    // Get Sunday by adding 6 days to Monday
    LocalDate sunday = monday.plusDays(6);

    // 3. Define the query to get the data
    String query = "SELECT day, COUNT(*)  FROM num_steps " +
            "WHERE day BETWEEN \"" + monday.format(formatter) +
            "\"  AND \"" + sunday.format(formatter) +
            "\" GROUP BY day ORDER BY day ASC";
    Cursor cursor = database.rawQuery(query, new String[] {});

    String tmpKey;
```

```
    Integer tmpValue = 0;

    // Define the week
    for (int i = 0; i < 7; i++){
        tmpKey = monday.plusDays(i).format(formatter);
        map.put(tmpKey, tmpValue);
    }


    // 4. Iterate over returned elements on the cursor
    cursor.moveToFirst();

    for (int i = 0; i < cursor.getCount(); i++){
        tmpKey = cursor.getString(0);
        tmpValue = Integer.parseInt(cursor.getString(1));

        //2. Put the data from the database into the map
        map.put(tmpKey, tmpValue);
        cursor.moveToNext();
    }

    // 5. Close the cursor and database
    cursor.close();
    database.close();

    // 6. Return the map with hours and number of steps
    return map;
}
```



3. Column chart

Create Column chart in Java using AnyChart library API.

DayFragment.java

```
public Cartesian createColumnChart(){
    //***** Read data from SQLiteDatabase *********/
    stepsByDay = StepAppOpenHelper.loadStepsByDay(getContext(),
current_time);

    //***** Create column chart using AnyChart library *********/
    Cartesian cartesian = AnyChart.column();

    List<DataEntry> data = new ArrayList<>();

    for (Map.Entry<String,Integer> entry : stepsByDay.entrySet())
        data.add(new ValueDataEntry(entry.getKey(), entry.getValue()));

    Column column = cartesian.column(data);

    //***** Modify the UI of the chart *********/
    column.fill("#1EB980");
    column.stroke("#1EB980");


    column.tooltip()
            .titleFormat("At day: {%X}")
            .format("{%Value} Steps")
            .anchor(Anchor.RIGHT_BOTTOM);

    column.tooltip()
            .position(Position.RIGHT_TOP)
            .offsetX(0d)
            .offsetY(5);

    // Modifying properties of cartesian
    cartesian.tooltip().positionMode(TooltipPositionMode.POINT);
    cartesian.interactivity().hoverMode(HoverMode.BY_X);
    cartesian.yScale().minimum(0);


    cartesian.yAxis(0).title("Number of steps");
    cartesian.xAxis(0).title("Day");
    cartesian.xAxis(0).labels().rotation(-45);
    cartesian.background().fill("#00000000");
    cartesian.animation(true);

    return cartesian;
}
```

Repository:

https://github.com/dadegrande99/AssignmentsMWC_Grandesso/tree/master/Assignment03