Mobile and Wearable Computing SA 2024-2025
Università della Svizzera italiana

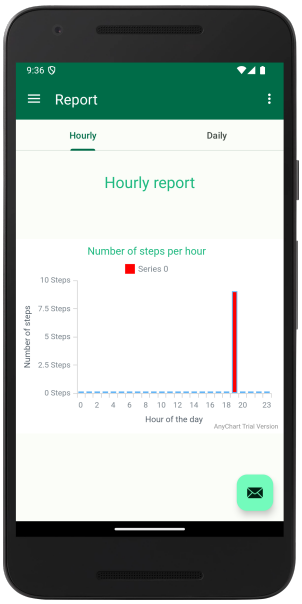
# Assignment 2

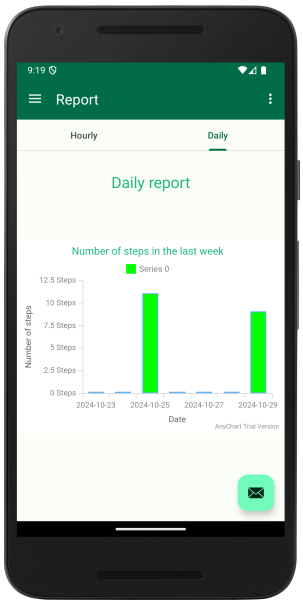
Paolo Deidda
paolo.deidda@usi.ch
https://github.com/USI-Projects-Collection/MWCTutorial05_Starter

October 30, 2024

## Contents





# 1 Data storage and visualization

As can be seen in Figure 1 the Report fragment has now a two switch tab that allows the user to select the time chunks for the chart in the same View.

## 1.1 fragment_report.xml

The first thing I needed to was to add was the **TabLayout** in file *fragment_report.xml*.

../app/src/main/res/layout/fragment_report.xml

```
18      <com.google.android.material.tabs.TabLayout
19          android:id="@+id/tabLayout"
20          android:layout_width="match_parent"
21          android:layout_height="wrap_content"
22          app:layout_constraintTop_toTopOf="parent"
23          app:layout_constraintStart_toStartOf="parent"
```

I renamed the **AnyChartView** from *hourBarChart* to *barChart* since it will now follow a double purpose; not only for hourly display but also for daily total count of steps.

## 1.2 StepAppOpenHelper.java

In this class, I implemented the method **loadStepsByDateForLastWeek** to retrieve the daily step counts for the past week from the database, which are then visualized in the report.

This method queries the database for each of the last seven days, counting the recorded steps for each day and storing the results in a `Map<String, Integer>`. The map pairs each date with its corresponding step count, allowing the app to display a daily summary in the report chart.

The implementation ensures that each date's steps are fetched correctly and displayed in order, by iterating over the last seven days and formatting dates according to the database's format. This is achieved using a `Calendar` instance and a `SimpleDateFormat` object, which matches the database date format (e.g., `"yyyy-MM-dd"`).

../app/src/main/java/com/example/stepappv4/StepAppOpenHelper.java

```
125
126     public static Map<String, Integer> loadStepsByDateForLastWeek(Context context) {
127         Map<String, Integer> stepsByDateMap = new TreeMap<>();
128
129         StepAppOpenHelper databaseHelper = new StepAppOpenHelper(context);
130         SQLiteDatabase database = databaseHelper.getReadableDatabase();
131
132         Calendar calendar = Calendar.getInstance();
133         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault())
                ;
134
135         for (int i = 0; i < 7; i++) {
136             String date = sdf.format(calendar.getTime());
137
138             String query = "SELECT COUNT(*) FROM " + TABLE_NAME + " WHERE " + KEY_DAY
                    + " = ?";
139             Cursor cursor = database.rawQuery(query, new String[]{date});
140
141             if (cursor.moveToFirst()) {
142                 int stepsCount = cursor.getInt(0);
143                 stepsByDateMap.put(date, stepsCount);
144             }
145
146             calendar.add(Calendar.DAY_OF_MONTH, -1);
147             cursor.close();
148         }
149
150         database.close();
151         return stepsByDateMap;
```

The `loadStepsByDateForLastWeek` method includes an SQL query to count the step entries recorded in the database for each date in the past week. The query works as follows:

- The SQL command `SELECT COUNT(*)` counts the total number of rows that match a specific condition, in this case, the number of entries for a given day.
- The table name `TABLE_NAME` is the name of the database table where step data is stored, while `KEY_DAY` represents the column containing each entry's date.
- The `WHERE` clause specifies a condition, with `KEY_DAY = ?`, where `?` is a placeholder that gets replaced by a date parameter.
- The parameter `date` is inserted into the query using the `rawQuery` method's second argument, which passes an array of parameters to replace placeholders in the query string.

After executing the query, a `Cursor` object is used to retrieve the results. If the query successfully finds entries for the specified date, the number of steps is extracted using `cursor.getInt(0)`, which accesses the first column in the result set, representing the step count.

For example, if `TABLE_NAME` is `"steps"` and `KEY_DAY` is `"date"`, with `date = "2024-10-28"`, the query would look like this:

```
1           SELECT COUNT(*) FROM steps WHERE date = "2024-10-28";
```

This query counts all rows where the `date` column matches `"2024-10-28"`, giving the total number of steps recorded for that day. The resulting count is stored in the `stepsByDateMap` for later visualization in the chart.

## 1.3 ReportFragment.java

This class has been heavily refactored as it had a long method called *createColumnChart* that was previously used to generate the hourly chart. Key parts of this method has been isolated to be reused also for the generation of the daily chart.

../app/src/main/java/com/example/stepappv4/ui/Report/ReportFragment.java

```
130     @Override
131     public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
            {
132         super.onViewCreated(view, savedInstanceState);
133 
134         TabLayout tabLayout = view.findViewById(R.id.tabLayout);
135         tabLayout.addTab(tabLayout.newTab().setText("Hourly"));
136         tabLayout.addTab(tabLayout.newTab().setText("Daily"));
137 
138         tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
139             @Override
140             public void onTabSelected(TabLayout.Tab tab) {
141                 if (tab.getPosition() == 0) {
142                     binding.textView5.setText(R.string.hourly_report);
143                     hourlyChart.setVisibility(View.VISIBLE);
144                     dailyChart.setVisibility(View.INVISIBLE);
145                     Toast.makeText(getActivity(), "Hourly Report Selected", Toast.
                            LENGTH_SHORT).show();
146                 } else {
147                     binding.textView5.setText(R.string.daily_report);
148                     hourlyChart.setVisibility(View.INVISIBLE);
149                     dailyChart.setVisibility(View.VISIBLE);
150                     Toast.makeText(getActivity(), "Daily Report Selected", Toast.
                            LENGTH_SHORT).show();
151                 }
152             }
153 
154             @Override
155             public void onTabUnselected(TabLayout.Tab tab) {}
156 
157             @Override
158             public void onTabReselected(TabLayout.Tab tab) {}
159         });
160     }
```

The main method is displayed in Code 3. This method handles the logic to switch between the two charts. Unfortunately **anyChart** class does not collaborate and does not reset the chart once created. Extensive trouble-shooting and many different attempts and approaches to make it work had been taken without success. The last attempt which seemed the most promising had the goal to create two charts and switch their visibility based on an event listener on the two Tab's buttons. In order to achieve to the *fragment_report.xml* I added another *AnyChartView* Object as shown in Code 4.

../app/src/main/res/layout/fragment_report.xml

```
20      <com.anychart.AnyChartView
21          android:id="@+id/hourlyChart"
22          android:layout_width="match_parent"
23          android:layout_height="300dp"
24          app:layout_constraintBottom_toBottomOf="parent"
25          app:layout_constraintEnd_toEndOf="parent"
26          app:layout_constraintHorizontal_bias="0.0"
27          app:layout_constraintStart_toStartOf="parent"
28          app:layout_constraintTop_toBottomOf="@id/tabLayout"
29          app:layout_constraintVertical_bias="0.501"/>
30
31      <com.anychart.AnyChartView
32          android:id="@+id/dailyChart"
33          android:layout_width="match_parent"
34          android:layout_height="300dp"
35          app:layout_constraintBottom_toBottomOf="parent"
36          app:layout_constraintEnd_toEndOf="parent"
37          app:layout_constraintHorizontal_bias="0.0"
38          app:layout_constraintStart_toStartOf="parent"
39          app:layout_constraintTop_toBottomOf="@id/tabLayout"
40          app:layout_constraintVertical_bias="0.501"/>
```

In method *onViewCreated* 3 I simply handle their visibilty based on the current selected Tab.

## 2 Important Notes

The problem has been investigated with both TAs but we could not figure out the exact source of the problem. A fast internet research suggest that this might have to do with the fact that anyChart is a free trial version and does not provide graph refreshing.