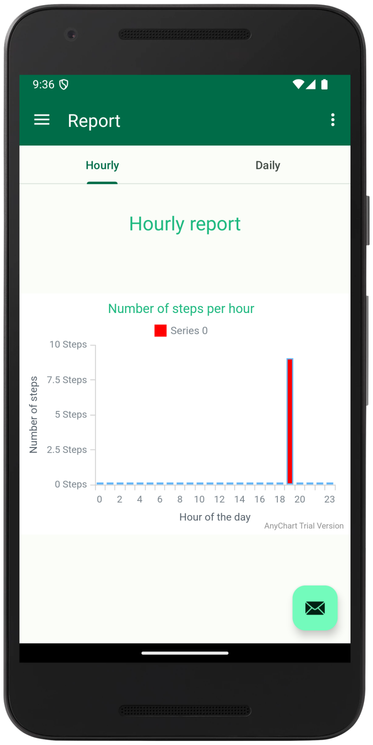
Mobile and Wearable Computing SA 2024-2025
Università della Svizzera italiana

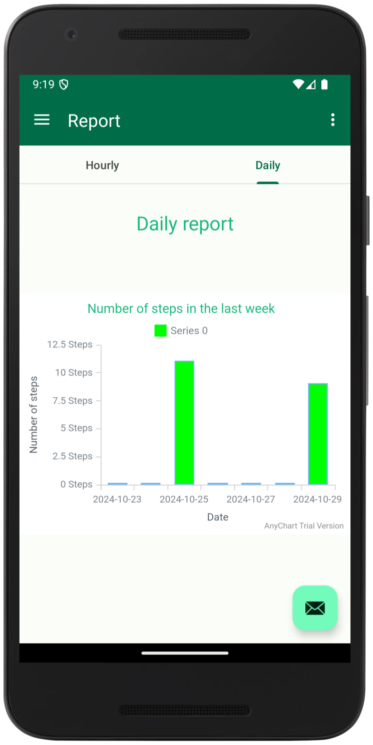
Assignment 2

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

```
    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="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

```
    public static Map<String, Integer> loadStepsByDateForLastWeek(Context context) {
        Map<String, Integer> stepsByDateMap = new TreeMap<>();

        StepAppOpenHelper databaseHelper = new StepAppOpenHelper(context);
        SQLiteDatabase database = databaseHelper.getReadableDatabase();

        Calendar calendar = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault());

        for (int i = 0; i < 7; i++) {
            String date = sdf.format(calendar.getTime());

            String query = "SELECT COUNT(*) FROM " + TABLE_NAME + " WHERE " + KEY_DAY + " = ?";
            Cursor cursor = database.rawQuery(query, new String[]{date});

            if (cursor.moveToFirst()) {
                int stepsCount = cursor.getInt(0);
                stepsByDateMap.put(date, stepsCount);
            }

            calendar.add(Calendar.DAY_OF_MONTH, -1);
            cursor.close();
        }

        database.close();
        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:

```sql
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

```java
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        TabLayout tabLayout = view.findViewById(R.id.tabLayout);
        tabLayout.addTab(tabLayout.newTab().setText("Hourly"));
        tabLayout.addTab(tabLayout.newTab().setText("Daily"));

        tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
            @Override
            public void onTabSelected(TabLayout.Tab tab) {

                if (tab.getPosition() == 0) {
                    binding.textView5.setText("Hourly report");
                    anyChartView.setChart(createHourBarChart());
                    Toast.makeText(getActivity(), "Hourly Report Selected", Toast.LENGTH_SHORT).show();
                } else {
                    binding.textView5.setText("Daily report");
                    anyChartView.setChart(createWeeklyBarChart());
                    Toast.makeText(getActivity(), "Daily Report Selected", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onTabUnselected(TabLayout.Tab tab) {}

            @Override
            public void onTabReselected(TabLayout.Tab tab) {
                if (tab.getPosition() == 0) {
                    anyChartView.setChart(createHourBarChart());
                } else {
                    anyChartView.setChart(createWeeklyBarChart());
                }
            }
        });
    }
```