LAM Project 2023

Federico Montori, Luca Sciullo

April 2023

Rules

In this document we describe one possible project for the exam of "Laboratorio di applicazioni mobili" course. The project can be implemented by **groups of maximum 2 people** (individual projects are obviously allowed as well), provided they implement the project integration. Each student/group can choose to develop the project proposed here (valid until February 2024) or suggest something else based on his/her personal interests. In the latter case, project proposals **must be individual only** and should be submitted via e-mail to Dr. Luca Sciullo (luca.sciullo@unibo.it), with a brief description of the application goals, contents and a list of **all** the features that you propose to implement. In extremely exceptional cases we can allow group projects outside the one proposed here, but they will need to be discussed with us in advance. The use of Git (or any other versioning system) is strongly encouraged. The following project description contains a minimum set of requirements for the application. Students are strongly encouraged to expand the track, by adding new features to the application, and/or further customizing the contents. Any question regarding the projects is to be asked to Dr. Luca Sciullo via e-mail (`luca.sciullo@unibo.it`). Regardless of the choice, every student/group is required to produce:

1. *One or more mobile applications*, there is no constraint on the language and the framework used: it can be native or hybrid, but it cannot be a Web Application (*i.e.* it must run natively on the device).

2. *A project report*, which is a document that describes the application produced, focusing on the workflow and the design choices. In particular, the Report should be named `SURNAME1_SURNAME2.pdf` and contain:

    - The name, surname, email, and matriculation number of each component of the group.
    - Overview of the application with screenshots.
    - Implementation details on how you chose to implement the functionalities.

    A good report is probably between 10 and 15 pages. Less than 5 pages are probably bad, and more than 15 is probably too many. The quality of the report WILL be part of the evaluation. **A good report also contains about 70% of implementation details and choices and only the remaining 30% of screenshots, overview...**

3. *A presentation*, which consists of a set of slides that will help you in the project discussion. They should contain a brief recap of the report alongside screenshots of the application. We suggest producing around 10 - 15 slides since the discussion time is approximately 15 minutes (a group discussion may last longer). Furthermore, **each** component of the group must know the details of the entire implementation and will be possibly asked about it during the project discussion.

The **CODE** of the mobile application and the **REPORT** have to be uploaded exclusively on the Virtuale platform in the dedicated section[1] (there are 6 deadlines throughout the year). They must be enclosed in a single .zip file named `SURNAME1_SURNAME2.zip`. In the case of a group, a single component is in charge of completing the upload. If the archive is too big for Virtuale, you can remove the directory "build" from your code (Android only). If the archive is still too big, then the student(s) must share the zipped code via Drive/OneDrive, etc. In this case, the students must still upload on Virtuale a zip file containing the Report, the link to the shared code, and its **hash**, so that we can prove that it was delivered on time. Projects delivered via e-mail will **NOT** be taken into account. The SLIDES, instead, must be brought along on the day of the oral examination.

Do not forget that the oral examination consists also in a theoretical part, which is **individual**. Therefore, knowledge of the topics discussed in class (both iOS and Android) is required. The exam consists of the project discussion (which is per group) and the oral examination (which is per student) and **must** be booked via Almaesami.

The Project: "Cellular Connectivity and Noise Map"

Overview

In the following project, the student is required to implement an interactive application that monitors the connectivity and the acoustic noise within geographical areas and colors the map consequentially to obtain a discrete heatmap. In particular, the app MUST implement the features listed below to be sufficient. Adding features is strongly encouraged and has good repercussions in the evaluation. A project that follows the specifications to its minimum but has no extra feature cannot reach high grades.

[1] `https://virtuale.unibo.it/course/view.php?id=28374`

Monitor the cellular connectivity and the acoustic noise

The app should be able to monitor the signal strength received by (1) the LTE or UMTS network interface, (2) the WiFi signal strength, and (3) the acoustic noise measured by the microphone and store such data locally for analysis. Each observation should result in a number (dB) and, for all three types of measurements, the student should identify a scale with at least 3 different ranges identifying, for instance, the classes “poor”, “average” and “good” (for the signal strength) or “loud”, “average” and “quiet” (for the sound).

Perform measurements in non-overlpping areas

The application has to map the position of the user as well as any data collected in the geographic map. As the user moves in the map (when the application is active) a value has to be produced by the application and it has to be reported on the map in the form of a color associated **with the area the user is in**. Google Maps uses the GPS coordinates to encode locations, however, the student must shift from a location (point) based encoding to an area encoding. The student can choose the way he/she wants to encode areas, with the following two constraints:

- the whole map can be covered (e.g. if we use non-overlapping circular areas there will unavoidably be “holes” in the map).
- the areas must not overlap.

hint: one way to do it could be using GPS-dependent square or hexagonal areas, or MGRS coordinates.

Each of these areas should be colored in a different way on the map according to the measurement taken (signal strength or sound). The coloring can be of any type, although it should be clear what colors stand for (e.g. red for poor, yellow for average, green for good). Therefore the app should be able to display 3 different maps (or overlays over a single map): one for the LTE signal strength, one for the WiFi signal strength, and one for the acoustic intensity.

MGRS MGRS (`https://en.wikipedia.org/wiki/Military_Grid_Reference_System`) is a way to divide uniquely the world map into squares. It is a hierarchical system, so we have a unique division for different sizes of squares (spanning from 1 meter to 100 km). In our case, we only want to use the squares with a side of 10 meters, 100 meters, and 1 km. A possible expansion of the project would be to use areas of different sizes to show the map at different granularities.

Encode the color

Each area should be colored according to the average of the last X measurements (X can be set by the user) and each measurement in the same area must be taken at least n minutes after the previous one (n can be set by the user). If the project is expanded with areas of different sizes, then obviously, a single measurement taken for a "small" area (say, a 10 m MGRS square) counts also for the bigger square that contains it (say, a 100 m MGRS square).

Notification and Storage

Measurements have to be taken either actively (i.e. the user presses a button that triggers the measurement) or in background (selectable in the app settings), in any case, there has to be a notification system that warns the user when he or she is in a zone where no new measurements have been taken for the current time frame (e.g. the present day). The developer can add filters to the notification system making it less "pedantic" (e.g. notify only for the squares of 100 m side or bigger).

Project Integration for 2 people

In the case of a project for 2 people, students must implement also the following additional feature: the application should be able to merge data coming from other users. More in detail, the application has to store, process, and visualize the other users' data, while keeping each data source separate in memory. For the first two tasks, we suggest using different databases/tables, while for the last one, it is possible to show the data use using different map layers, different colors, etc... The only requirement here is that the application must be able to show both only the local data or all merged data on request. Data should be merged according to the position coordinates (using the average), but more sophisticated algorithms can be used as well and will contribute to the final evaluation of the project. Users can share their data in different ways:

- *Remote cloud sync.* Users can use a cloud-shared database to synchronize their data, like Firebase. The sync action can be triggered manually or automatically following custom strategies, and the data schema used must be customized by the students. This does not replace the local database for his/her own data.
- *Import/Export of a database dump, using a custom data format.* Users can share the file containing their data through an external app and import it into the application to process the additional data. The schema used for exporting the data must be defined by the students.
- *Proximity share.* Users can share their data when approaching a proximity area with another user. The proximity area can be created using, for instance, BLE beacons. If two users become visible in the same area, they are notified that someone else is available and they can use an M2M communication (WIFI-direct, Bluetooth) for sharing their data. The data schema used must be customized by the students and the range of the area can be customized through the application settings.

All the previous strategies are valid for the exam evaluation, but (from the top to the bottom) the difficulty increases and so does their weight for the final evaluation.