Android Project Proposals

Federico Montori, Luca Sciullo

21st April 2020

Abstract

In this document, we describe two possible projects for the exam of **"Laboratorio di applicazioni mobili"** course. Each student can choose a project from the set, or suggest something else based on his/her personal interests. In this latter case, project proposals should be submitted via e-mail to Dr. Luca Sciullo (*luca.sciullo@unibo.it*), with a brief description of the application goals, contents and requirements. The following project descriptions should be considered as hints. Students are strongly encouraged to expand the tracks, adding new features to the applications, and/or further customizing the contents.

Any question regarding the projects have to be asked to Dr. Luca Sciullo via e-mail.

The CODE, the REPORT of the projects have to be uploaded exclusively on the IOL platform in the dedicated section. The SLIDES, instead, must be brought the day of the oral examination.

The Report should contain:

- Overview of the application with screenshots.
- Implementation details on how you chose to implement the functionalities

A good report is probably between 8 and 15 pages. Less than 5 pages is probably bad, more than 20 is probably too much. The quality of the report WILL be part of the evaluation. The slides will help you during the project discussion and should contain a brief recap of the report alongside with screenshots of the application. We suggest to write max. 10 slides since the discussion time is approximately 10 minutes.

1 Personal health monitor

In the following project the student is required to implement an interactive application for tracking personal information about his/her health to be saved into daily reports. In particular, the application should be able to manage the reports within a calendar, send notifications and plot the data according to specific filters.

1.1 Create and modify reports

The application must be able to create, modify and delete health reports. Health reports are summaries of health information tracked by the user that should be saved every time the user thinks to be a good idea, and at least once per day. Each report must include a minimum number of two information related to the user's health (e.g., body temperature, blood pressure, Glycemic index, etc..). Each information has an importance, i.e., an index that specifies the level of attention that requires such parameter (from 1 to 5), and each report has an optional note that can be filled with auxiliary information. Reports are stored by the application (we strongly suggest to use a database) and there must be the possibility to show reports on a daily basis, like for instance inside a calendar. In case of multiple reports for the same day, a summary report should be created, where the health information is the average of all collected data of that day. There has also to be the possibility to visualize reports according to some filters (e.g., only the reports with importance set to 5).

1.2 Notifications

The application must notify the user if he/she has not inserted yet a report for that day. In this case, the user can perform the following actions: postpone the reminder (in such case the user will be asked for another time and date of the same day) or open directly from inside the notification the form for filling the report. The time in which the notification is sent by the application can be set by the user from a settings page. The application must also notify the user if the average of data collected for an information - with importance greater than 3 - in a fixed period of time has exceeded a predefined threshold. User can customize the previous parameters from a settings page, i.e., he/she can decide which information should be monitored, for how long time and what is the threshold that must not be reached.

1.3 Graphs

The application should be able to collect usage stats displaying at least two graphs of any kind (pie graph, box plot, histogram, line plot, etc.) showing useful data (e.g. the variation of an health information over a week, the variation of the number of collected reports each day, etc..).

2 GiftToMe

GiftToMe is a web platform for sharing old stuff that can be gifted to people who need such things. GiftToMe is particularly useful to avoid wastes and in those situations where shops and markets will remain closed for a long time. The idea is to gift something - still working and in good conditions – that we do not consider useful anymore for ourselves but that could be useful for someone else.

In the following project the student is required to implement an interactive client application for posting, displaying and managing information on the GiftToMe platform. For the sake of simplicity, each new article is posted on Twitter according to a specific format. The list of available articles can be retrieved by looking for specific posts on Twitter. Interactions among users are also possible by posting replies on Twitter.

2.1 Create, modify, delete and visualize posts

The application must be able to create, modify and delete posts on GiftToMe platform, i.e., on Twitter. The post must follow the following format:

```
#LAM_giftToMe_2020-article
{
  "id": "9a61bc0c-ce39-4d3b-b72d-e63af675522a",
  "issuer": "username",
  "category": "sport",
  "name": "Racchetta da tennis",
  "lat" : -37.835309,
  "lon" : 145.047363,
  "description":
    "Ottima racchetta usata solo un paio di volte."
}
```

where the first line identifies the Tweet typology and:

- id [UUID]: it is the unique identifier of the post
- issuer [string]: it is the twitter nickname of the person who posts the article
- category [string]: it is the category the article belongs to. There are 5 different categories: sport, electronics, home&office, toys, other
- name [string]: it is the name of the article
- lat [float]: it is the latitude where the article is located
- lon [float]: it is the longitude where the article is located
- description [string]: it is an arbitrary string to describe the article

2.1.1 Create a post

The application must provide a form for filling the information required for a new article and it must be able to post the new article on Twitter.

2.1.2 Modify a post

The application must offer the possibility to modify some information of a post that was previously posted on twitter by the user. In particular, an user has to be able to modify the name, category, position (i.e., lat and lon), and description of an article that he/she owns.

2.1.3 Delete post

The application must offer the possibility to delete an old post. This event should also occur when an article is gifted to someone else, hence it is not available anymore.

2.1.4 Visualize posts

The application must offer the possibility to visualize all the available posts. Some filters should be implemented to reduce the list (e.g., category filter).

2.2 Manage replies to an existing post

The application must be able to create, modify, delete and visualize replies to already existing posts. Replies are normal tweets with the following format:

```
#LAM_giftToMe_2020-reply
{
  "id": "04367a30-e487-439d-8f10-3f891f0883cc",
  "sender": "senderUsername",
  "target": "9a61bc0c-ce39-4d3b-b72d-e63af675522a",
  "receiver": "username",
  "message": "Ciao, sono interessato."
}
```

where:

- id [UUID]: it is the unique identifier of the reply
- sender [string]: it is the twitter nickname of the person who replies to a post
- target [UUID]: it is the unique identifier of the article that is the subject of the reply
- receiver [string]: it is the twitter nickname of the person the reply is addressed to

- message [string]: it is an arbitrary string containing the message of the reply

2.2.1 Create a reply

The application must provide a form for filling the information required for a new reply and it must be able to post the reply on Twitter.

2.2.2 Modify a reply

The application must offer the possibility to modify the message field of a reply that was previously posted on twitter by the user.

2.2.3 Delete a reply

The application must offer the possibility to delete an old reply. This event should also occur if the original post containing the target of the reply (the target field) is not found anymore on giftToMe.

2.2.4 Chat dashboard

The application must provide a simple but effective interface for visualizing replies to an owned article . As optional task, the application can provide an interface for keeping going the conversation with other users.

2.3 Notifications and geofancing

The user should be notified if there are available articles close to him. In particular, a notification should be sent to the user each time a new article becomes available in a position inside a radius of 1km, where the center is represented by the user's position. Furthermore, the application must notify the user if he/she has received a reply to a post (only the ones containing an article) that previously published on giftToMe.

3 Alternative projects

We here propose harder and bigger tracks for projects to be done in conjunction with internships/thesis. These topics are strongly focused on research. Further information can be asked via email to the tutor.

3.1 Web of Things

"The Internet of Things (IoT) is widely recognised to have lots of potential, but its commercial potential is being held back by fragmentation. A sensor on its own has limited value, but there are huge opportunities for open markets of services that combine sensors, actuators and multiple sources of information.

The Web of Things seeks to counter the fragmentation of the IoT, making it much easier to create applications without the need to master the disparate variety of IoT technologies and standards. Digital twins for sensors, actuators and information services are exposed to consuming applications as local software objects with properties, actions and events, independently of the physical location of devices or the protocols used to access them."

Projects in the context of W3C Web of Things are mainly about building dashboards for displaying, managing and enabling the mashup of data coming by heterogeneous Web Things.

3.2 Mobile Crowdsensing

Mobile Crowdsensing uses smartphones of a big crowd to monitor phenomena of common interests happening in the real world (mostly cities but also nature). This can happen participatorily or opportunistically. Projects in this area involve the development of clients for MCS and need a decent set of skills in either C++, Python or simulation.

3.3 Service Computing for IoT

Projects in this are a involve skills in service computing, service compositions and data analysis for IoT. The projects will involve data gathering from heterogeneous open data sources, data integration, data classification and the dynamic service composition through smart user interfaces.