

Android project proposals

Luca Bedogni Marco Di Felice ({lbedogni,difelice}@cs.unibo.it)

April 19, 2015

Introduction

In this document, we describe three possible projects for the exam of the "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 Luca Bedogni (lbedogni@cs.unibo.it) and/or to Marco Di Felice (difelice@cs.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.

1 Wireless Hot Spot finder

This project proposal consists in an application that should discover wireless access points at the current user's location, save their data on an internal database and display them on Map. In particular, the application should:

- Discover access points in the current range.
- Visualize discovered access points on a map.
- Color the areas on which wifi coverage might be present

In the following, each features is further discussed and analysed in detail.

1.1 Find access points in range

The application should listen to the 802.11 channels to spot the presence of wifi networks in the current user's range, and save them on a internal database together with:

- Location where the access point has been found
- Signal strength
- Security type (WEP, WPA, Open ecc.)
- ... (any further information available on the access point)

1.2 Plot found access points on a map

The application should be able to visualize the discovered access points on a map, identifying them with different icons and colors. In particular, open access points should be displayed as **green**, secured access points should be displayed as **red**. Moreover, developers should foresee different visualization methods (e.g. visualize all the access points, visualize only the access point in a given range, visualize only the open access points, etc), that might be selected by the end-user.

1.3 Color the areas on which wifi coverage might be present

The application, based on its internal database, should color the map where radio coverage might be present. To implement this functionality, triangulation algorithms could be used to determine the coverage area of each access point.

2 Mail client

This project consists in an e-mail client for the Android Mobile platform. The client should be able to:

- Configure email accounts.
- React on mailto intents.
- Send/Receive/Forward/Delete emails.

In the following, each feature is further discussed and analysed in detail.

2.1 Configure email accounts

The applications should be able to add a new user account, configuring the username, the password and the server related information. Considerations should be taken into account about the storage of username and password, and appropriate security mechanisms should be used.

2.2 React on mailto intents

When the user clicks on a mailto link, the application should offer to send an email to the address contained in the link. Appropriate Intent filters should be configured in order to offer this functionality.

2.3 Send/Receive/Forward/Delete email

The applications should be able to perform classical email operations, such as sending/receiving/forwarding/etc. The remote server could use POP3 or IMAP, based on user preferences. Moreover, message filters and folders might be implemented (optional).

3 Mobile Latex editor

LaTeX is a language to write documents, particularly used for scientific publications and reports. Via a LaTeX compiler, it is possible to generate PDF files from a LaTeX source file. However, the LaTeX compiler requires considerable resources in terms of memory and computation resources, and thus it might not be supported by a mobile device.

The project consist in developing a LaTeX editor for the Android platform, that is able to compile the document on a remote server, and then open the corresponding PDF on the device in use. More in detail, the app should:

- Provide a text editor for managing text files.
- Support the LaTeX syntax.
- Foresee the possibility to compile the LaTeX source code on an external server, and then download the resulting PDF.

3.1 Provide a text editor for managing text files

The application should provide classical functionalities of a text editor, i.e. the possibility to open, edit, save, close a file.

3.2 Support the LaTeX syntax

The application should be able to help the user in writing LaTeX code. Syntax highlighting should be provided. Moreover, the applications should provide facilities to insert LaTeX code (e.g. helping the user to insert math symbols).

3.3 Remote compiling

When the user clicks on the Compile button, the file should be automatically transfered to a remote server, where a PDF compiler is working. Once the PDF has been generated, it should be transferred back to the mobile applications. In this case, an Intent should be generated to open the PDF file. Optional elements:

- Handle the compiler logs (e.g transfer them back to the mobile applications)
- Manage .tex files with images. In this case, the app should be able to compress the folder in which the source files are (i.e. tex files and images), and send it to a remote server. At server-side, the compressed archive should be extracted, compiled, and the corresponding PDF should be returned back to the device. After that, the extracted directory on the remote server should be deleted.

4 Bologna OpenData

The Bologna Municipality has made available several data in an open format, through the website <http://dati.comune.bologna.it/>. These include more than 800 different dataset concerning the mobility, municipal buildings, and many more.

The goal of the project is to leverage on the data provided to offer different kind of services to the user. The application should be able to update the data by using the permalink API offered by the municipality, the details of which are published on the specific dataset to be used.

This project does not have specific guidelines, as everything depends on the chosen dataset. Thus, students are advised to check with Dr. Luca Bedogni and/or Dr. Marco di Felice before starting to develop their own project.

5 Project bonus

While for the exam it is sufficient to implement the requirements presented above, we strongly encourage to enrich the application by adding new original features to the applications.

6 Projects submission

Projects must be submitted through email to `lam-projects@cs.unibo.it` and should also be uploaded on the Wilma-Lab wiki (<http://www.cs.unibo.it/projects/wilma-lab/wiki/doku.php?id=start>). In this case, students must firstly ask for a wiki account via email to Luca Bedogni., and then provide a brief description of the project (the description could also be a part of the final project report) in their personal wiki page.