



ft_hangouts

Le Grand Ambassadeur

Résumé: This project is designed to get you acquainted with the android system. You will have to create a contact management application.

Table des matières

I	Preamble	2
II	Introduction	3
III	Goals	4
IV	General instructions	5
V	Mandatory part	6
VI	Bonus part	7
VII	Emulation	8
VIII	Turn-in and peer-evaluation	9

Chapitre I

Preamble

No snow on Mount Fuji,
Nore flowers in the cherry trees,
No grass on my flowerbeds,
its reign is mineral.

Magic is the liquid,
magic is the source,
magic is the crystal,
magic is the garden.

[illegible]

crystal, crystal, crystal, crystal
 You cannot touch it.
 You're not ready, you can just gaze it,
 Your eyes are ready, you can stand it.
 Wooaaaa
 Oh it's beautiful, it's beautiful, it's beautiful

The wizard gardener
doen't need manure in his garden.

Shiny like stars,
Shiny like a waveless sea,
Shiny like a big dog teeth.

Shiny like insects,
Shiny like a sausage,
Shiny like teen skin.

crystal oh crystal
crystal oh crystal
crystal oh crystal
crystal oh crya ha ha

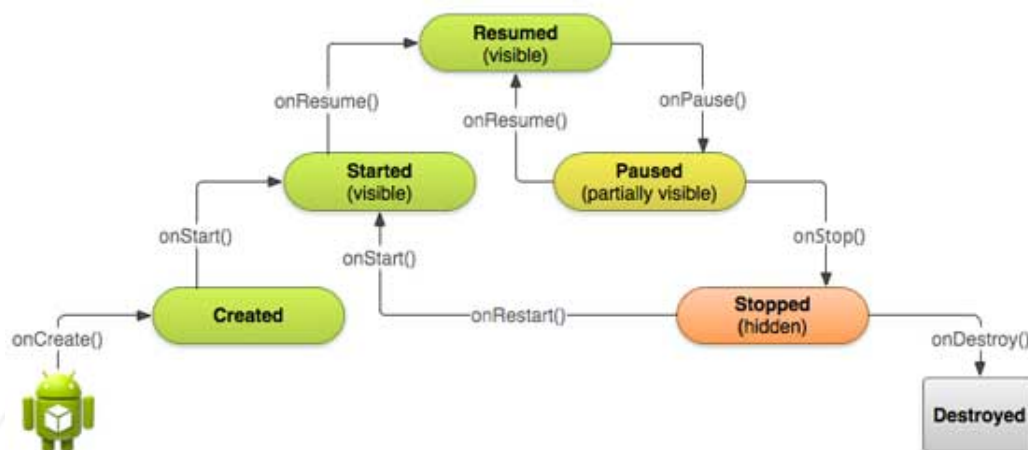
Salut C'est Cool - *The wizard gardener* [lien](#).

Chapitre II

Introduction

For this project, you will have to design an android application that will allow to create a contact and send texts.

The goal is to understand how an Android application works, how Android deals with you app and how to use the SDK.



Chapitre III

Goals

You will have to fulfill various tasks that will help you understand how an Android app in JAVA or Kotlin works. The goal is to make an app allowing to create a contact (containing at least 5 details), edit it and delete it. Once the contact is recorded, you will have to be able to communicate with him through text messages.

Contacts are recorded persistently (SQLite database, don't use the shared contact table. You must create your own). A summary for each contact will appear as a list on the app's homepage. You should be able to click on each contact to show their details.

Your app will have to propose two different languages, one being the default language (change the system language for a test). When you're on the homepage and set the app in the background, the date will be saved and will show in a *toast* when you return to the front. You will have to create a menu that will allow you to change the app's header color. Finally, the app icon will have to be the 42 logo.

Chapitre IV

General instructions

- This project will be evaluated by humans only.
- The project will have to be coded in JAVA or Kotlin
- No external library is allowed (even for design).



It is strongly advised that you use Android Studio as your IDE.
Beware, Google does not support the ADT plugin for Eclipse anymore.

Chapitre V

Mandatory part

Here is what you will have to create :

- Create a contact.
- Edit a contact.
- Delete a contact.
- Homepage with a summary for each contact.
- Receive text messages from recorded contacts.
- Send text messages to your contacts.
- Create a menu that will allow you to change the header color.
- The app will have to include two different languages.
- Show the time the app was set in the background when returning to the app.
- The app works in landscape and portrait modes.
- The app icon is the 42 logo.

Chapitre VI

Bonus part

- Have a picture for each contact.
- When receiving a text message, a new contact is instantly created with the number as the contact name.
- It's beautiful! Material Design is cool!
- You can call the contact.

Enjoy yourself, there are many ways you can improve your app!

Chapitre VII

Emulation

This part will explain how to use an Android emulator.



The emulator included in Android Studio does not yet work with dumps. You should use Android Studio on a VM or a real device if you want to test the text messages.

You can find many Android emulators and their quality is uneven : [GenyMotion](#) is very effective and easy to install (VirtualBox is required). However, its trial version doesn't allow to send text messages.

The AVD (Android Virtual Device) included in Android Studio is not the most effective, but you can control it. It is installed with Android Studio and is quite intuitive.

To send a text message from the AVD :

```
student@e1rip1$ telnet localhost {port, (window title)}
#Trying 127.0.0.1...
#Connected to localhost.
#Escape character is '^]'.
#Android Console: type 'help' for a list of commands
#OK
sms send {number} {message}
```

For a call :

```
student@e1rip1$ telnet localhost {port, (window title)}
#[...]
gsm call {number}
```

While an image is being executed, the computer sees it as a phone. Simply choose the image when launching the app from Android Studio.

Chapitre VIII

Turn-in and peer-evaluation

Beware when you turn in your test. Many files, more or less useful, are generated in your project. Configure your `.gitignore` [hint](#) properly.

For the grading, the project will be compiled and installed with :

```
./gradlew installDebug
```