ft_hangouts

The Great Ambassador

Pierre-Elie Kesslassy pkesslas@student.42.fr

Summary: The goal of this project is to make you familiar with Android system by creation a contact management application.

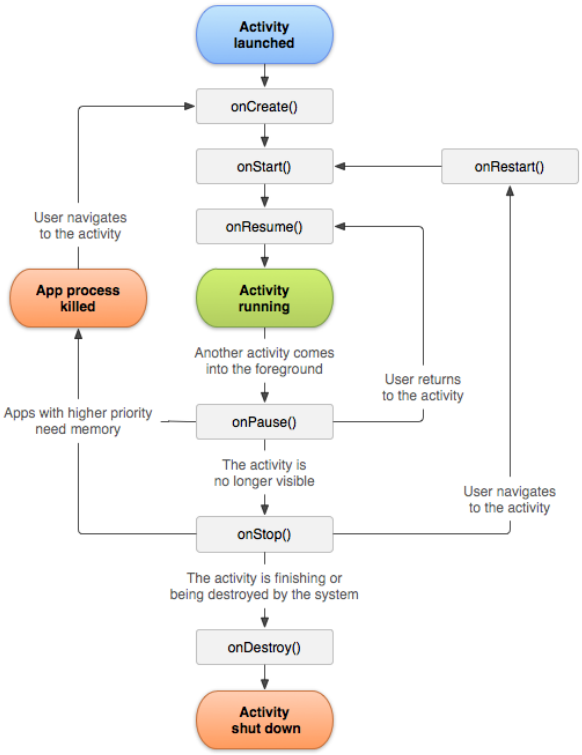
Contents

Chapter 1

Introduction

In this project, you will need to make an Android application that will manage contacts and allow you to chat with them via SMS.

The goal of this project is to understand how an Android application works, how Android manages your application and how to use the SDK.



Chapter 2

Objectives

You will have to perform various tasks that will make you understand how an Android application in JAVA or Kotlin. The goal is to make an application that allows to create a contact (with at least 5 information field), to edit it and to delete it. Once registered contact it will be possible to start dialogue with him via SMS.

Contacts must be persistently saved (in SQLite database, do not use the shared contact table, but create your own). A summary of each contact must be present on the application home list, a click on a summary must show all the information about the corresponding contact.

Your application will also have to work in two languages, one of which is default (change the language of the system to test). When you are on the home page and you set the application in the background, the date must be saved and displayed in a toast when you return app to the foreground. You must be able to change the color of the header of the application from preferences menu. And finally, the icon of the application must be the logo of 42.

Chapter 3

General Instructions

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

It is strongly recommended to use Android Studio as IDE. Be careful, the ADT plugin for Eclipse is no longer supported by Google.

Chapter 4

Mandatory Part

Here's what you will need to do:

- Creation of the contact.
- Edition of the contact.
- Deletion of the contact.
- Homepage with a summary of each contact.
- Receiving SMS from saved contacts.
- Sending SMS to your contacts.
- Changing header color from menu.
- Support of two languages.
- Display the time of setting app to the background when you return to the application.
- Landscape and portrait mode support.
- The logo of the application must be 42.

Chapter 5

Bonus Part

Bonuses will only be counted if your mandatory part is PERFECT. By PERFECT, we obviously mean that it is fully realized, and it is not possible to alter its behavior in default, even in case of error, misuse, etc ... Concretely, this means that if your mandatory part is not validated, your bonuses will be fully IGNORED.

- Photo for contacts.
- On sending an SMS to the new number, a contact with the number in name is directly created.
- It's beautiful! Material Design is cool. (Without external libs such as "AppCompat" etc...).
- Abillity to make call to the contact.

Be creative, lots of things can improve the application.

Chapter 6

Emulation

This part explains how to use an Android emulator.

> The emulator provided with Android Studio does not work for the moment. I recommended you to use Android Studio on a VM or a real device if you want to test SMS

Several Android emulators exist, some more or less good: GenyMotion is very powerful and easy to set up (VirtualBox is required) but does not allow sending SMS in its free version.The AVD (Android Virtual Device) that comes with Android Studio is not the most powerful, but it gives more control. It is set up with Android Studio, and it's intuitive.

To send an SMS from the AVD:

```
gopa@e1r1p1$ 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 {numero} {message}
```

For a call:

```
gopa@e1r1p1$ telnet localhost {port, (window title)}
#[...]
gsm call {nubmer}
```

When an image is running, the computer recognizes it as a phone. Choose that image when you launch the application from Android Studio.

Chapter 7

Correction and peer-evaluation

Make your work on your GiT repository as usual. Only the present work on your repository will be assessed in defense.

Beware of your repository, many more or less useful files are generated in your project. Remember to set up your .gitignore hint.

For the correction the project will be compiled and installed with:

```
$./gradlew installDebug
```