WebSem Lab Report

Android Implicit Intents

Name of Student 1: Zachari Thiry
Name of Student 2: Matthieu RAMON
Promo: 2023

MobServ : Mobile Applications and Services
(Fall, 2022)

EURECOM Graduate School and Research Center in Digital Science
Data Science Cursus

October 20, 2022

Abstract

This lab consists in an introduction to application development on Android, and to the notion of implicit intents.

Contents

1 Exercises

1.1 When and for what purpose are the following methods called during the lifecycle of an Activity?

- **onCreate() :** This method is called when the Activity enters the *"Created"* state. The commands containted in this function are generally used to initialise the variables, configure the listeners, initiate the connections and set up the graphical interface. After its execution, the app isn't yet visible nor interactive.
- **onStart() :** This method is called when the Activity enters the *"Started"* state. After its execution, the user interface becomes visible but the user can not yet interact with it.
- **onResume() :** This method is called when the Activity enters the *"Resumed"* state. After its execution, the app is expected to be fully operational and will remain in such a state until it is interrupted by an other event (call, popup...)
- **onDestroy() :** This method is called when the Activity is stopped, when the user calls finish() or the system decides to free allocated resources.
- **onStop() :** This method is called when the Activity enters the *"Stopped"* state, for example when an other activity is called on top of it. This method is generally used to store some data about the running Activity so that it may be resumed on the same state;
- **onPause() :** This method is called when the Activity enters the *"Paused"* state. It can be considered as the opposite of the onResume() method. Everything that is started by onResume() must be put on hold in the onPause() method. onPause() is generally used when the activity is still visible, but not on the foreground (eg. pop up window partially covering the screen). In this method, optimization is of the essence.

1.2 In the ImplicitIntent example, could we use a different OnClick function per button?

It is indeed possible to use different functions for OnClick. To do that, one should simply create a new function alongsise ImplicitIntent with a different name (eg. ImplicitIntent2) and specify in the button that this method should be call. However, it can be considered as bad practice in some cases since this inevitably leads to code repetition and clumsiness.

1.3 Use an example to illustrate the relationship between different application components.

Using the dial button example :

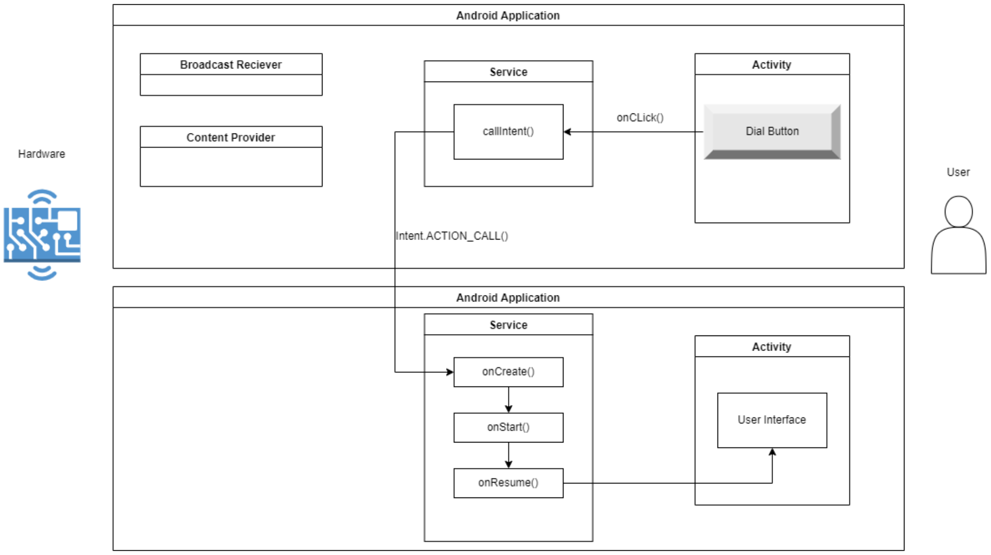

Figure 1: Accuracy of models given different inputs

1.4 Show an app chooser based on the instructions online:

See built application, on button *"Chooser Button"*

1.5 Add another button that, when pressed, adds events to the calendar.

See built application.