Philip Tham
Mikael Ragnhult

TIG055
S14

Report for Android project - Counter with NFC features

Basic Features of the Application

This is a report about the Android counter application developed for this course. The basic features of the application are incrementing and decrementing by using different methods and tools. Incrementation and decrementation can be done in three ways: using the touch buttons displayed on the main activity, using the volume keys on the Android device, as well as scanning NFC tags.

The layout is different in Portrait mode and Landscape mode. In Portrait the increment and decrement buttons are at the bottom of the activity display while in Landscape mode the buttons are to the right for the sake of ease of access.

The settings allow the user to turn the sounds on and off when buttons are pressed and enable/disable the use of the volume keys for counting.

This application has the possibility of using NFC tags to increment and decrement. By reading a specific string from a tag the application can react to the intent and perform the appropriate action. A future feature of the application is to be able to write strings to NFC-tags so that different counters can make use of unique tags.

The use of a database has been implemented in the code. If the database does not contain an appropriate table the application will send information to the database for creation of a new one. Otherwise, if the corresponding table already exists, the application will get the values needed for its counters and start.

Tools

The application was coded in Eclipse for Android SDK, and tested on physical Android devices such as the Samsung Galaxy S5 and Google Nexus 7 (2012), as well as the Android Virtual Device provided by Eclipse.

The use of physical Android devices proved to be much faster than emulating the operating system in the virtual device. This also helped with testing the hardware buttons, vibration, etc.

Developer Documentation

The main documentation used as assistance was the Android Developers training site, as it provided the majority of the information needed to develop this application.

Android Platform

Issues came up during debugging, as the error messages were illegible when compared to standard Java exceptions. This made it difficult to find the source of the bugs. As a getaround toast messages were used in combination with Exception.getMessage() to find the reason to the problem. There were also problems when the system wanted to recreate the state when rotating the display in various situations. For example when a dialog is open and text is written to a text field inside the dialog the information is lost when the screen is rotated.

There were also some problems getting used to the use of fragments at the beginning of development. There were some difficulties in designing the user interface with fragments, as they would not align properly.

Since there are so many different versions of Android, screen sizes, and processors, applications must be developed and adapted with these in mind.

One positive thing about developing for Android is that common files, such as preferences, are automatically managed and are easy to use.

New Experiences Discovered through Android Programming

Programming with Android proved to be an educative experience, as it allowed us, as developers, to learn how applications work in the background, such as how activities are terminated whenever the display shifts from Portrait mode to Landscape mode and vice versa.

The separation of user interface in XML code and actual functionality in Java code is a new and smart way of applying the MVC model in coding.

Development Progress

The application started out with a single counter that could increment and decrement. The settings allowed the user to set maximum and minimum boundaries, and the amount that the buttons would increment/decrement. All the values were stored in the local Shared Preferences. As development of the application advanced, the possibility to create and use multiple counters was implemented, where it was possible to keep different values in different counters.

Misc.

As the application is not yet in its final state, the following features will be implemented as future projects.

- When creating a new counter, the device will be able to write a unique id to an NFC tag
- Counters will be able to react specifically to their respective tag
- The application will react to a tag that is for a counter stored in the device and switch over to the appropriate counter
- With the use of Android Beam, users will be able to exchange counters with each other
- Create counters that can only increment on specific events, such as specific tags and only once
- Setting maximum and minimum values for counters (already implemented, except for user interface)
- Categories in navigation drawer
- Expand the usage of the database to contain unique NFC IDs