

Android Application „LectureInfo“

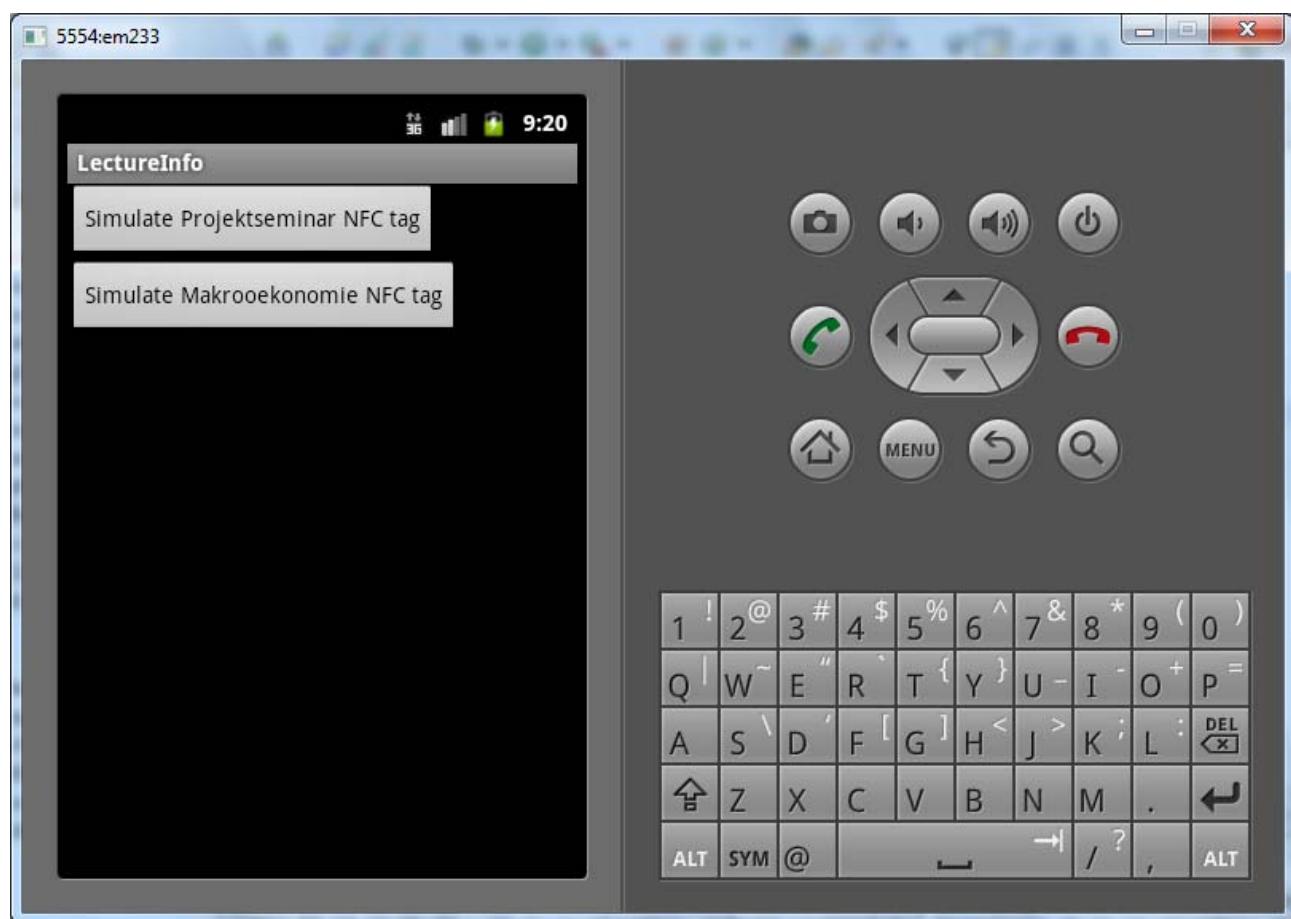
Für das Projekt ist die Guava-Library erforderlich. Diese muss heruntergeladen und im Build Path des Projekts eingebunden werden! (siehe Ende des Dokuments)

Die nachfolgende Abbildung zeigt den Opening Screen der LectureInfo App. Normalerweise ist ein solcher gar nicht vorgesehen, denn das Programm bestünde nur aus einer Activity, die sich als *Broadcast Receiver* registriert und wartet, bis vom System der folgende Broadcast gesendet wird: `android.nfc.action.TAG_DISCOVERED`

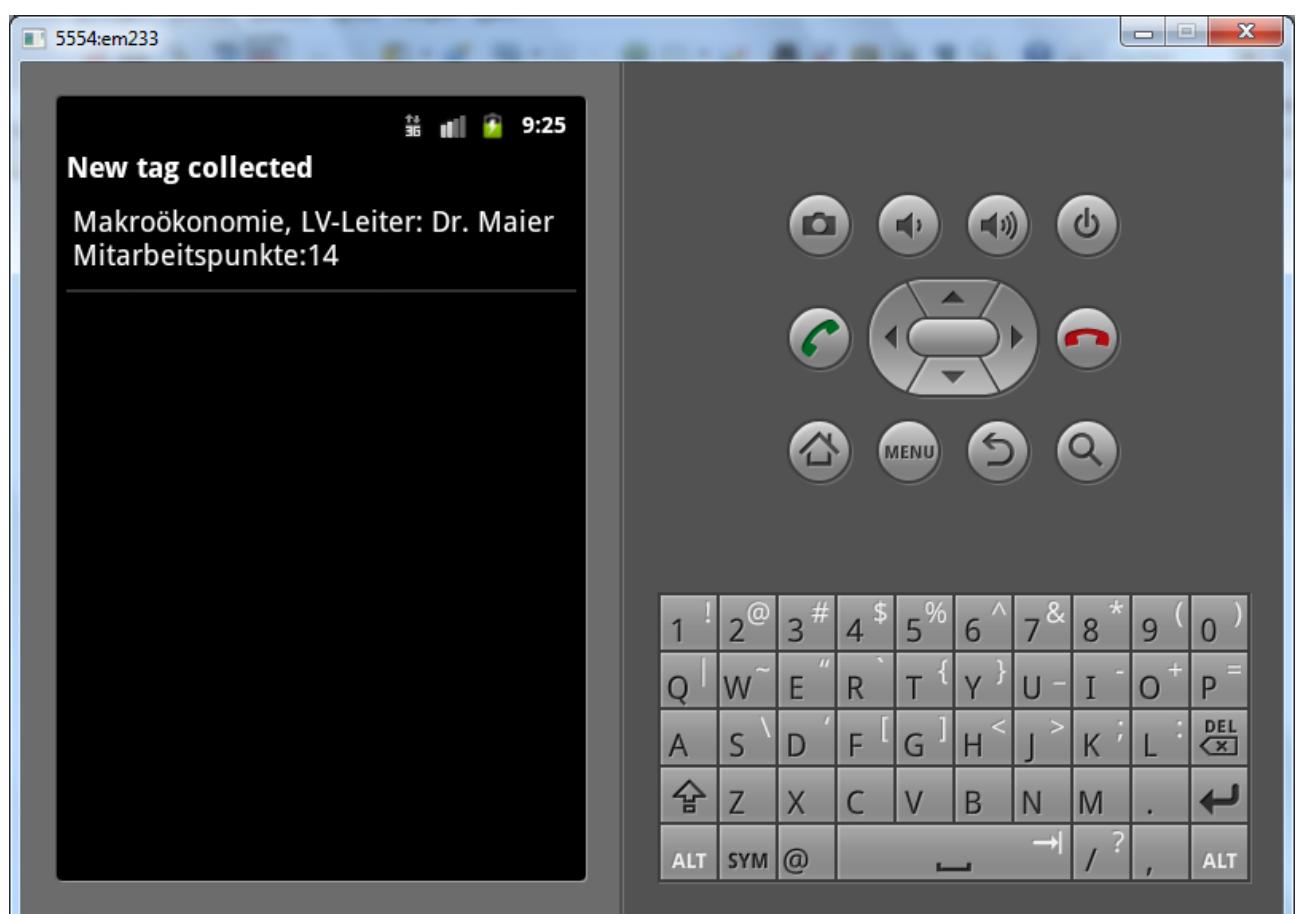
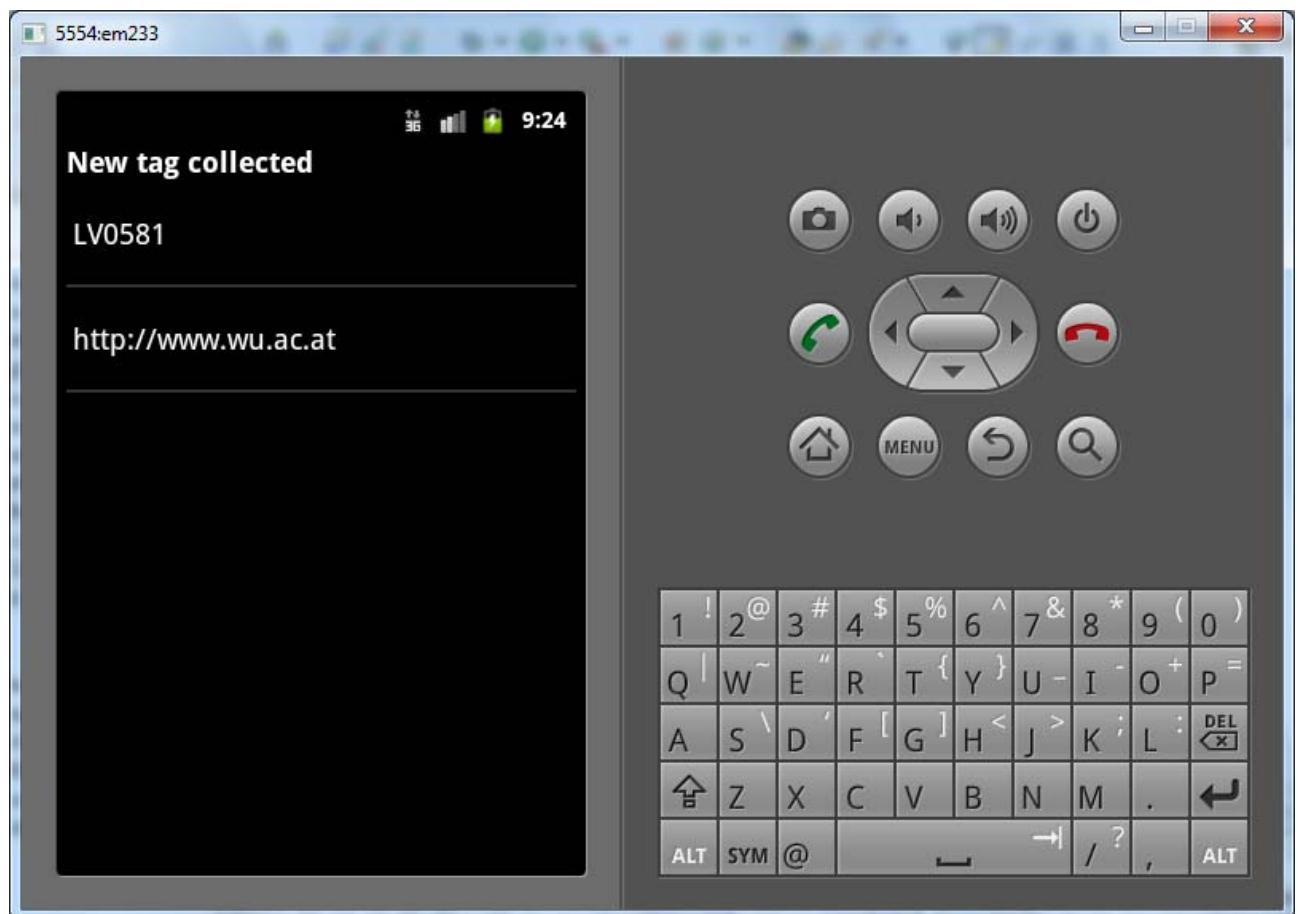
Dann würde einfach die TagViewer Activity ausgeführt, welche vom NfcAdapter die NFCMessage bezieht, parst und graphisch darstellt.

Da allerdings keine NFC-Hardware und kein programmierbares NFC Terminal bei der Entwicklung zur Verfügung stand, musste der Prozess simuliert werden. Dies kommt im Opening Screen zum Ausdruck, der über zwei Buttons verfügt. Diese erzeugen „virtuell“ NdefRecord Objekte (NDEF: NFC Data Exchange Format) und in Folge NdefMessages, welche an den NfcAdapter gesendet werden, wie dies wohl von der Treiberseite der Hardware passieren würde.

Man kann sich also vorstellen, dass die Betätigung eines Buttons wie das Heranhalten an ein NFC Terminal wirkt (welches vor dem jeweiligen Raum montiert ist, in dem das Projektseminar oder Makroökonomie stattfindet).



Wird ein Button betätigt, geschieht das oben geschilderte und die TagViewer Activity wird nicht direkt, sondern über einen Broadcast gestartet.



Bei Projektseminar (LV Nr. 0581) sind in der NdefMessage zwei NdefRecords enthalten: Der Name und eine URL.

Zum Quelltext: Durch Betätigung des Buttons wird ein byte-Array erzeugt, welches den NdefRecord darstellt (siehe NFC Spezifikationen, wie dieser aufgebaut sein muss). Es handelt sich um ASCII-Hexcodes (Mapping z.B. hier: <http://unicode.org/Public/MAPPINGS/ISO8859/8859-1.TXT>). Dieser NdefRecord wird dann in eine NdefMessage verpackt.

```
case R.id.projektseminar:  
  
    byte[] tag = new byte[] { (byte) 0xd1, (byte) 0x02, (byte) 0x1c,  
                            (byte) 0x53, (byte) 0x70, (byte) 0x91, (byte) 0x01,  
                            (byte) 0x09, (byte) 0x54, (byte) 0x02, (byte) 0x65,  
                            (byte) 0x6e, (byte) 0x4c, (byte) 0x56, (byte) 0x30,  
                            (byte) 0x35, (byte) 0x38, (byte) 0x31, (byte) 0x51,  
                            (byte) 0x01, (byte) 0x0b, (byte) 0x55, (byte) 0x01,  
                            (byte) 0x77, (byte) 0x75, (byte) 0x2e, (byte) 0x61,  
                            (byte) 0x63, (byte) 0x2e, (byte) 0x61, (byte) 0x74,  
                            (byte) 0x00, (byte) 0x00 };  
  
    NdefMessage[] message;  
    try {  
        message = new NdefMessage[] { new NdefMessage(tag) };  
    } catch (FormatException e) {  
        throw new RuntimeException("NdefMessage could not be created", e);  
    }  
  
    final Intent intent1 = new Intent(NfcAdapter.ACTION_TAG_DISCOVERED);  
    intent1.putExtra(NfcAdapter.EXTRA_NDEF_MESSAGES, message);  
    startActivity(intent1);  
  
    return;
```

Über einen Intent wird dem System dann mitgeteilt, dass ein NFC Tag identifiziert wurde (dies würde, vermute ich, normalerweise die Implementierung des Treibers der NFC-Hardware machen). Die NdefMessage wird dem Intent mitgegeben. Dann übernimmt das Android System die Entscheidung, was passieren soll, und da sich die TagViewer Activity als Broadcast Receiver registriert hat, wird diese gestartet, um sich um die Aktion zu kümmern (Parsing und Darstellung des Tags).

Bei Makroökonomie bei Dr. Maier ist ausschließlich ein Text als einziger NdefRecord vorhanden, welcher wieder in einer NdefMessage gekapselt ist, und dieser wird angezeigt. Der relevante Quelltext sieht folgendermaßen aus:

```
case R.id.macroeconomics:  
    int score = 14;  
    String lectureInfo = "Makroökonomie, LV-Leiter: Dr. Maier\n"  
                        + "Mitarbeitspunkte:" + score;  
  
    NdefRecord[] rec = { newTextRecord(lectureInfo, new Locale("de"),  
                           true) };  
    NdefMessage[] msg = new NdefMessage[] { new NdefMessage(rec) };  
  
    final Intent intent2 = new Intent(NfcAdapter.ACTION_TAG_DISCOVERED);  
    intent2.putExtra(NfcAdapter.EXTRA_NDEF_MESSAGES, msg);  
    startActivity(intent2);
```

Ein String enthält die relevanten Infos (man könnte hier zB auch eine Datenbankabfrage einbauen). Mithilfe der Funktion newTextRecord wird uns die Arbeit des manuellen Erzeugens eines byte-Arrays abgenommen (diese Funktion stammt aus dem NFCDemo der Android Developers – siehe ZIP und Projektdateien).

Hinweise

Die TagViewer Activity und die Parser im Projekt wurden 1 zu 1 vom NFCDemo übernommen!

Die Klassen sind hier dokumentiert:

<http://developer.android.com/resources/samples/NFCDemo/index.html>

Guava-Library Download: <http://code.google.com/p/guava-libraries/>

In Eclipse: Rechtsklick auf das Projektverzeichnis → Properties → Java Build Path:

