**NFC FEATURE IMPLEMENTATION LOGIC**

**N/B:** Before performing any read or write operations, you must authenticate the tag block sections part,

nfcA.authenticateSectorWithKeyB(2, MifareClassic.KEY\_DEFAULT)

this block I authenticated, convers from block 7-11 I think

**WRITE**

* The Nfc tag must be have the techlist = [NfcA, miraClassic, Ndefformatable]
* I only write to block 8 and block 9 of the tag
* if the input string length is less than 16, I write to only block 8 and clear block 9
* else if greater than 16, I split the input string by the substring
  + first substring is from (0,15) , second substring is (15, $length of input string)
  + since nfc tag only supports byte array, I convert the string to byte array, using Utf8

val dataToSend: ByteArray =  
 *userEmail*!!.toByteArray(StandardCharsets.UTF\_8)

* + after converting both substrings to byte array, then I store the first part in block 8, and then store the second part in block 9

**READ**

* The Nfc tag returns the data as byte array, so I read from block 8 and block 9, as those are the block we are writing to.
* After getting the byte array from this block, we check if block 9 is empty and if its empty byte array we just return an empty string,
* Then we convert the byte array returned by block 8 to string using Utf\_8
* If block 9 byte array isn’t empty, convert it to string using Utf\_8 also, and add it to the string gotten from block 8 so as to have one string,
* The block 8 string should come first,
* Now since the tag has other empty blocks, it returns “?” along with the read result, so just replace “?” with “” after joining the strings of both blocks ,