

**ÇANKAYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING**

MASTER THESIS

**DEVELOPMENT OF NFC APPLICATIONS:
USE CASE FOR SOCIAL NETWORKS**

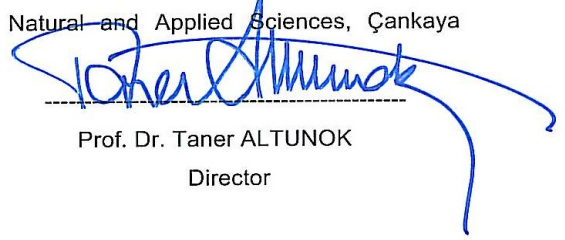
SERKAN ÖZKAN

SEPTEMBER, 2012

Title of the Thesis : **Development Of NFC Applications: Use Case For Social Networks**

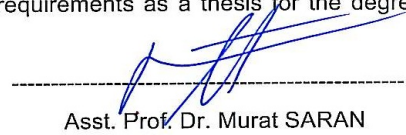
Submitted by : **Serkan Özkan**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University:



Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Murat SARAN
Computer Engineering Department Acting Head

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Murat SARAN
Supervisor

Examination Date : 17.09.2012

Examining Committee Members:

Asst.Prof.Dr. Murat SARAN (Çankaya University)

Asst.Prof.Dr. Abdül Kadir GÖRÜR (Çankaya University)


Asst.Prof.Dr. Erol ÖZÇELİK (Atılım University)



STATEMENT OF NONPLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Serkan Özkan

Signature : 

Date : 17.09.2012

ABSTRACT

DEVELOPMENT OF NFC APPLICATIONS: USE CASE FOR SOCIAL NETWORKS

Özkan, Serkan

M.S.c., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Murat Saran

September 2012, 77 Pages

This study introduces Near Field Communication (NFC) technology, analyzes its current state and tries to form a use case for the application of NFC technology for social networking. The present study focuses on Android platform, Android NFC support, and how Android Application Programming Interfaces (APIs) can be used to develop a sample NFC aware social networking application. The study also briefly analyzes social networks and mobile devices, both of which are now undeniable parts of our lives. Sample use cases are built around a restaurant scenario in which the restaurant tries to make more customers, who are at the restaurant in person, participate in social networking actions related to the restaurant. Sample scenarios describe how NFC technology and mobile devices can help generate more publicity and revenue by simplifying customer-vendor interactions. This study has showed that NFC technology and NFC enabled Android devices can be used to simplify and improve social networking interactions.

Keywords : NFC, Android, Social Network, Social Action, Mobile, Ndef, Smart Poster

ÖZ

NFC UYGULAMALARININ GELİŞTİRİLMESİ: SOSYAL AĞ KULLANIM SENARYOSU

Özkan, Serkan

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Murat Saran

Eylül 2012, 77 Sayfa

Bu çalışma Near Field Communication (NFC) teknolojisinin güncel durumunu analiz ederek bu teknolojinin sosyal ağlar için kullanımını içeren bir kullanım senaryosunu ortaya koymayı amaçlamaktadır. Bu çalışma Android platformu, Android NFC desteği ve Android Uygulama Programı Arabiriminin (UPA) örnek sosyal ağ kullanım senaryoları için nasıl kullanılacaklarını incelemektedir. Bu çalışma günlük hayatımızın değişmez birer parçası haline gelmiş olan sosyal ağlar ve mobil cihazları da kısaca analiz etmektedir. Çalışmada uygulanan örnek kullanım senaryoları restorandaki müşterilerin restoran ile ilgili sosyal ağ etkileşimlerinde bulunmalarını amaçlayan bir restoran etrafında kurgulanmıştır. Örnek senaryolar NFC teknolojisinin ve mobil cihazların iş yeri-müşteri ilişkisini kolaylaştırarak nasıl sosyal ağlar üzerinden daha fazla tanıtım ve dolayısıyla daha fazla gelir elde etmek için kullanılacağını ortaya koymaktadır. Bu çalışma sonucunda NFC teknolojisi ve NFC desteğine sahip Android cihazların sosyal ağ etkileşimlerini basitleştirmek ve geliştirmek için kullanılacakları ortaya konulmuştur.

Anahtar kelimeler : NFC, Android, Sosyal Ağ, Sosyal Etkileşim, Mobil, Ndef, Smart Poster

ACKNOWLEDGMENTS

The author would like to thank to the Turkish Government and the Parliament for giving a second chance to complete his master's degree by granting an academic amnesty.

The author would also like to thank Asst. Prof. Dr. Murat Saran for his support.

TABLE OF CONTENTS

STATEMENT OF NONPLAGIARISM.....	iii
ABSTRACT	iv
ÖZ	v
ACKNOWLEDGMENTS.....	vi
CHAPTERS.....	1
INTRODUCTION.....	1
ORGANIZATION OF THE THESIS	1
BACKGROUND OF THE STUDY.....	1
STATEMENT OF THE PROBLEM	2
SIGNIFICANCE OF THE STUDY.....	2
SCOPE AND LIMITATIONS	3
INTRODUCTION	3
CHAPTER I NEAR FIELD COMMUNICATION - NFC	5
1.1. BACKGROUND.....	5
1.1.1. Historical Development	6
1.1.2. Future of NFC	6
1.1.3. ISO/ECMA Standards	7
1.1.4. Compatibility.....	7
1.1.5. Security	7
1.2. NFC FORUM SPECIFICATIONS	8
1.2.1. NDEF	8
1.2.2. NFC Forum Smart Poster Record Type.....	10

1.2.2.1. Smart Poster Use Cases by NFC Forum	12
1.3. USES OF NFC.....	14
1.4. NFC SUPPORT IN MOBILEDEVICES	15
1.4.1.1. Hardware Requirements.....	15
CHAPTER II SOCIAL NETWORKS	18
2.1. SOCIAL NETWORKS IN GENERAL	18
2.2. SOCIAL NETWORKS IN A MOBILE CONTEXT	19
2.3. FACEBOOK.....	21
2.3.1. Facebook NFC Use Cases	21
2.3.1.1. Use Case 1 – Basic Usage:.....	22
2.4. TWITTER.....	22
2.4.1. Twitter Use Case.....	23
2.4.1.1. Twitter Use Case 1 – Follow.....	23
2.4.1.2. Twitter Use Case 2 – Tweet	24
CHAPTER III ANDROID PLATFORM AND NFC ON ANDROID	25
3.1. ANDROID PLATFORM.....	25
3.1.1. Android Intent Mechanism	26
3.2. ANDROID NFC SUPPORT	27
3.2.1. Tag Technologies As Used In Android API.....	28
3.2.2. android.nfc.NfcAdapter	28
3.2.2.1. Intents.....	29
3.2.3. Permissions.....	30
3.2.4. android.nfc.NdefMessage	31
3.2.5. android.nfc.NdefRecord	32
3.2.6. android.nfc.Tag	33
3.2.7. android.nfc.tech.Ndef	33
3.3. TAG DISPATCH	33

3.3.1. NFC Modes of Operation	34
CHAPTER IV PROPOSED NFC ENABLED SOCIAL NETWORKING SOLUTION	36
4.1.1. Solution From Social Networking And Android Platform Points Of View	36
4.1.2. Solution from NFC Point of View.....	38
4.1.3. Application Flow	39
4.1.4. Security Concerns Related to the Solution	40
4.1.5. Future Use Cases	41
CHAPTER V PREVIOUS RELATED WORK	43
5.1. WORK ON NFC AND SOCIAL NETWORKS	43
5.2. WORK ON ALTERNATIVE TECHNOLOGIES	44
5.3. WORK ON SOCIAL NETWORKS	46
CHAPTER VI APPLICATION DEVELOPMENT ENVIRONMENT	47
6.1. SOLUTION ARCHITECTURE & REQUIREMENTS	47
6.2. ANDROID SDK	47
6.2.1. Android SDK Installation	49
6.3. DEVELOPMENT ENVIRONMENT	50
6.3.1. Debugging.....	50
6.3.1.1. Virtual Device	50
6.3.1.2. Debug Configuration.....	51
6.3.2. Testing	51
6.3.2.1. Unit Testing	51
6.3.2.2. Functional Testing	52
6.3.2.3. Applications Required For Testing On Virtual Device	52
6.3.3. Testing Using Open-NFC	53
6.3.4. Testing On Hardware Devices	53
CHAPTER VII THE APPLICATION	54

7.1. PACKAGE STRUCTURE	54
7.2. APPLICATION CONFIGURATION	55
7.3. COMPONENTS USED	55
7.4. IMPORTANT CLASSES AND METHODS	56
7.5. USING OFFICIAL NATIVE APPLICATIONS FOR SOCIAL NETWORKS .	56
7.5.1. Custom URL Schemes.....	58
7.6. USING A COMMON WEB PAGE	59
7.7. USING SOCIAL NETWORK SDKS	59
7.7.1. Facebook SDK	60
7.8. APPLICATION SCREENSHOTS.....	61
CONCLUSIONS AND DISCUSSIONS.....	65
RECOMMENDATIONS FOR FURTHER IMPROVEMENTS OF NFC TECHNOLOGY USE CASE FOR SOCIAL NETWORKING.....	66
RECOMMENDATIONS FOR FURTHER IMPROVEMENTS OF THE APPLICATION	67
REFERENCES.....	68
APPENDICES	75
Appendix A – SocialTouch Class Diagram	75
Appendix B – SocialTouch Use Cases	76
Appendix C – CURRICULUM VITAE.....	77

LIST OF TABLES

Table 3.1 US smartphone platforms distribution by comScore research [3].	25
Table 5.1 QR code – NFC comparison table by Vasquez-Brienso et al [63]	45

LIST OF FIGURES

Figure 1.1 : NDEF record structure	9
Figure 1.2 Sample smart poster definition taken from Smart Poster specification [16]	12
Figure 1.3 NFC Forum logo.....	13
Figure 1.4 A smart poster in action	13
Figure 1.5 NFC Uses cases defined by NFC Forum [7, 17].....	14
Figure 1.6 NFC antenna embedded in back cover of Huawei sonic, aka Turkcell T20.....	16
Figure 1.7 An SD card with embedded NFC antenna.....	16
Figure 2.1 %Who Currently Have a Personal Profile Page Any Social Networking Website	20
Figure 2.2 % of Social Network Users Owning a Smart Phone	20
Figure 2.3 Distribution of Users by Device Used to Access Facebook	21
Figure 2.4 Customer physical interaction use case illustration	24
Figure 3.1 Ndef tag dispatch flow chart [33].....	34
Figure 4.1 Flow diagram for the application	40
Figure 6.1 Distribution of Android devices by versions (Data collected from Google play store access statistics) [52].	48
Figure 6.2 Android SDK manager GUI.....	49
Figure 6.3 Android virtual device configuration	51
Figure 7.1 Application properties.....	54
Figure 7.2 Application screenshot 1	61
Figure 7.3 Share link on Facebook	62
Figure 7.4 Application screenshot 2 – Tweet action.....	62
Figure 7.5 Tweeting.....	63
Figure 7.6 Application screenshot – Follow on Twitter.....	63
Figure 7.7 Following on Twitter	64
Figure 7.8 Application screenshot - Dynamic tag generation.....	64

CHAPTERS

INTRODUCTION

ORGANIZATION OF THE THESIS

This study aims to define a use case for Near Field Communication technology for social networking purposes by defining a simple interaction using mobile devices, NFC technology, and ubiquitous computing.

The structure of the thesis is as follows:

- Introduction
- NFC technology and NFC use cases
- Social networking
- Android platform
- Proposed solution
- Development of a sample application
- Results and Conclusions
- Recommendations for further improvements

BACKGROUND OF THE STUDY

Mark Weiser (1991) [1] had used the following phrases to define ubiquitous computing: *"the most profound technologies are those that disappear, they weave themselves into the fabric of everyday life until they are indistinguishable from it"* (p.94). Today, mobile devices and social networking sites definitely fit into this definition. They are simply undeniable parts of daily life. Social networking sites are also major advertising and publicity platforms. Simplifying user-social network interactions and guiding or manipulating this interaction may be a serious business enabler.

NFC technology is an emerging technology that can help improve physical-cyber space interactions by simplifying interactions and transferal of data between

physical devices. NFC technology may allow otherwise nonfunctional objects like tables, menus, posters etc. to be utilized in cyber interactions and improve ubiquitous computing experiences.

STATEMENT OF THE PROBLEM

It is an important business objective to simplify business to customer interactions and gain more customers and revenue. Businesses can leverage social networks to advertise and gain new customers or keep in touch with their customers in new ways that were not available one or two decades ago.

Keller [2] has defined this new era as:

Traditional approaches to branding that put emphasis on mass media techniques are less and less effective in a marketplace where customers have access to massive amounts of information about brands, products and companies, and in which social networks have, in some cases, supplanted brand networks.

Especially when the business has only limited and mostly physical interaction with the customer, the business has a very narrow window of opportunity to extend this interaction to the cyber space. A business like a restaurant has to connect/interact with its customers while they are eating at the restaurant, in a very simple “ubiquitous” way.

There has been some work related to this limited physical interaction use case especially using barcode or QR code applications, but those technologies are not flawless, they have their own limitations and issues like difficulty of use, more effort by the user required to complete the process. NFC technology may be utilized to improve these interactions.

SIGNIFICANCE OF THE STUDY

This study aims to define a scenario and a solution that will simplify business-customer relations in order to enhance ubiquitous computing experiences by defining a solution using mobile devices and NFC technology. This study may help widespread dissemination of NFC technology by providing new use cases for NFC technology. The output of this study is an analysis of NFC technology, potential problems and solutions, and a sample proof of concept application.

SCOPE AND LIMITATIONS

The study focuses on only the two most popular social networking sites, Facebook and Twitter. All possible use cases are not defined for these social networks, only a few sample use cases are defined.

A native application for Android platform is developed. Therefore, the sample application does not support other mobile device platforms such as Apple iOS. Use cases are limited to a business-customer interaction scenario, although there may be other possible use cases that can be defined for other purposes like user-to-user interactions or customer-to-business interactions etc.

Developed application is not a production ready fully featured application; it is a proof of concept application. All related features of the underlying operating system or technology may not be utilized in real life settings.

INTRODUCTION

NFC, Near Field Communication, is an advancement of RFID and contactless smart card technologies which is slowly making its way into mobile devices. With NFC support, mobile devices become capable of reading and writing NFC tags, acting as contactless smart cards and card readers or communicating with other NFC enabled mobile devices.

Smart phones and similar mobile devices are spreading and advancing at an incredible speed. Current mobile devices are now quite capable computers and they are becoming more and more NFC enabled. Smart phones with NFC technology allow us to consolidate existing technologies like smart cards, payment or ticketing systems etc. into our mobile devices. Besides, the distinctive features of NFC capable devices such as to be used for payments, ticketing and identity related purposes may allow users to interact with each other and social networks in new ways.

Nowadays, social networks are one of the greatest forces shaping the world. They are used by billions of people every day and they are used for anything from sharing a video with friends to organizing revolutions.

Social networks are one of the most important advertisements, publicity platforms at the moment. Almost every company, organization or even individual have social network profiles, and they are trying to have more followers on twitter, more likes on Facebook, have their tweets retweeted by others, become a trending topic etc. To

increase return of investment ratios and generate more revenue, it is vital to simplify these social interactions. In this study, we will investigate how NFC and mobile devices can be used for this purpose. If we can define use cases and develop applications that allow easier interaction with social networks using NFC technology, this may also be an important accelerator for widespread use of NFC technology.

Android is the leading mobile device operating system, platform. As of August 2012 more than half of the entire smart phones are running Android [3]. It is supported and used by several major hardware device vendors and mobile network operators. Android has built-in support for NFC since Android version 2.3, released in December 2010. Android devices with NFC hardware are becoming more common, a good example for this is the Samsung Galaxy S3 or the new Google Nexus. Android NFC API greatly simplifies NFC development.

CHAPTER I

NEAR FIELD COMMUNICATION - NFC

1.1. BACKGROUND

NFC is a set of short-range wireless technologies. NFC can be considered as an advancement of RFID (radio frequency identification) and smart card technologies [4].

Its range is approximately 4 centimeters or less, theoretically the range can be up to 20 cm but in practice it is 4 cm or less. NFC uses magnetic induction between two loop antennas. NFC uses globally available unlicensed band of 13.56Mhz [5].

NFC interaction involves an initiator and a target. The initiator generates an RF field which can power targets without own power supplies.

NFC tags are basically little chips with some data embedded in them. Passive tags do not require some kind of power or battery, so they can be attached to any item like posters, appliances, wall mounted items etc. Also both ends of the interaction can be active devices, in which case it is possible to create a two way communication and transmit data between devices.

Main difference between RFID and NFC is the fact that NFC communication can be two-way whereas RFID communication is one-way. NFC devices can both receive and send information.

NFC operates at slow speeds and ranges but does not require pairing or manual setup. So NFC communication can be established in an easy and fast manner. When compared to other wireless protocols like Bluetooth, NFC is very slow. Maximum data transfer speed of NFC is approximately 0.424Mbps. So NFC may not be suitable for transferring large amounts of data.

1.1.1. Historical Development

Development of NFC technology was initially started by Sony and Philips [6]. Mobile FeliCa card embedded into DoCoMo and Sony phones have been widely used for mobile payments in Japan for several years even before the NFC Forum initiative. (Although similar to NFC technology those proprietary technologies are not compatible with NFC standards).

NFC Forum, an organization that aims to standardize NFC technology and related efforts, was founded in 2004 by NXP, Sony and Nokia to advance the use of NFC in consumer electronics, mobile devices and PCs. Goals of NFC Forum are defined as [7]:

- Developing standards, technical specifications that define a modular architecture and interoperability for NFC devices and protocols.
- Encourage the development of products using NFC Forum specifications
- Ensure that products claiming NFC capabilities comply with NFC Forum specifications
- Educate consumers and enterprises about NFC

In June 2006, NFC Forum outlined the architecture for NFC technology, released their first technical specifications and announced their NFC Forum compatibility certification program [7]. So NFC can be considered as a young technology in its early stages (but it may also be considered as old in this era of super-fast technological changes). As of August 2012, NFC has not yet reached wide-spread use and it is not yet supported by many vendors.

Currently NFC is mostly considered as a platform for payment applications. Google wallet [8] and Turkcell cep-t cüzdan [9] are major examples of payment applications using NFC technology. These applications are actually contactless smart cards emulations. They allow users to load one or more credit card data into the phone and use the phone instead of smart cards for payments.

1.1.2. Future of NFC

The future of NFC depends on the number of NFC devices and promotion of uses of the technology for purposes other than payments. Social network integration may play a key role in the future of NFC. Use of technology for purposes like social networking may be the key to widespread adoption. More users may be more likely to use the technology for social networking purposes than for payment purposes.

Sony has recently announced TransferJet [10] technology which is also a similar wireless technology operating at close proximity but capable of transferring data at speeds up to 350Mbps. Development of alternative technologies may be a risk for the future of NFC technology. Since it has not reached widespread use yet it may be easier to entirely replace the protocol with an alternative protocol. So the next few years may be critical for the future of NFC.

Especially lack of NFC support in existing Apple products is a very critical negative factor for wide-spread use of NFC. If upcoming versions of Apple iPhone have NFC support that may be a critical turning-point for NFC. For example if a future version of iPhone supports NFC then the number of NFC enabled devices and NFC applications will increase dramatically in a very short period of time.

1.1.3. ISO/ECMA Standards

NFC is built on ISO 14443, RFID standard. NFC standards are NFC Interface and Protocol -1, ISO18092/ECMA-340, and NFC Interface and Protocol -2, ISO21481/ECMA-352.

ECMA-340/ISO18092 [5, 11] defines active and passive communication modes of NFC.

ECMA-352/ISO21481 [13, 12] defines detection and selection of communication mode.

1.1.4. Compatibility

NFC is compatible with existing RFID terminals, card readers. So NFC enabled devices can be used with existing terminals. This compatibility allows NFC enabled devices to easily replace contactless smart cards.

1.1.5. Security

NFC is inherently secure due to its close proximity requirement, but at the same time it is considered insecure since the standard does not define the security requirements. Security is mainly the responsibility of the application developer.

Security is very critical in payment applications or applications that process sensitive data, but since the scope of this project is limited to uses of NFC for social networking we can accept that security based on proximity and existing mobile device use cases is adequate for the purposes of this project. Since we will not be storing or transmitting sensitive data we do not need high security standards for the

purposes of this study. The device must be unlocked to activate NFC support so it is unlikely that a device can be used without the users consent.

1.2. NFC FORUM SPECIFICATIONS

1.2.1. NDEF

NDEF stands for “NFC Data Exchange Format” and NDEF specification was released by NFC Forum [7] in 2006 [14]. NDEF specification defines a common binary data format for NFC Forum compliant devices and tags. NDEF allows encapsulation of payloads of arbitrary type and size. Payloads are defined by type, length, and identifier. NDEF payloads can be nested or can contain chains of linked chunks of unknown length.

NDEF specification [14] only defines the binary data format and does not define any data exchange methods or any record types. NDEF is designed to allow application developers to encapsulate and take advantage of existing well known data formats like xml.

An NDEF message contains one or more NDEF records [15] each carrying data of arbitrary type and up to $2^{32}-1$ octets. NDEF records can be chunked in a way similar to the chunked encoding used in http. This allows encapsulation of dynamically generated data but a chunked record must be contained within a single NDEF message, chunks cannot span multiple NDEF messages.

Each NDEF record contains three parameters [15]:

- Payload length: length of payload
- Payload type: type of the payload can be URI, a mime type or an NFC specific data type. Format of the type field is defined using a Type Name Format (TNF).
- Payload identifier: optional parameter which can be an absolute or relative URL which can be used to cross-reference payloads.

NDEF record layout is defined by NDEF specification [14] as seen in the Figure 2.1:

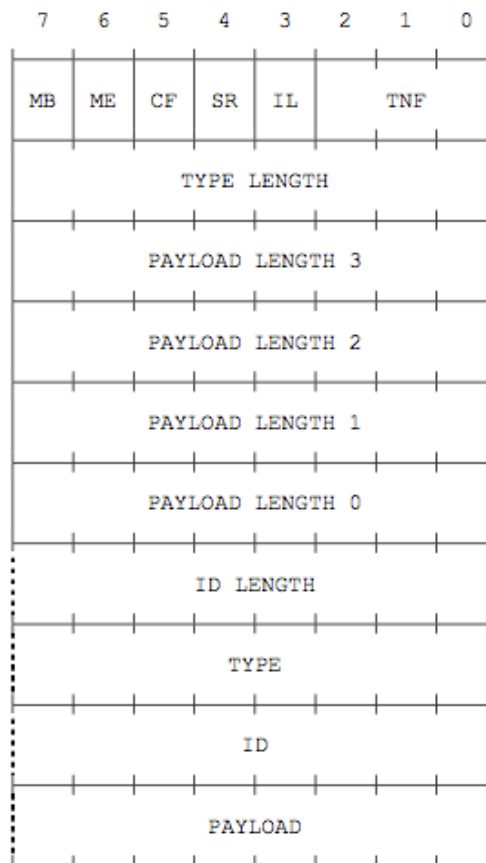


Figure 1.1 : NDEF record structure

In this figure:

- MB: Message begin bit flag
- ME: Message end bit flag
- CF: Begin/continuation chunk indicator bit flag
- SR: Indicator for the length of the PAYLOAD_LENGTH field
- IL: Bit flag that indicates if ID LENGTH field exists or not
- TNF: Type name format field, can be one of the following :
 - o 0x00 : Empty
 - o 0x01 : NFC Forum well known type
 - o 0x02 : RFC 2046 media type
 - o 0x03 : Absolute URI as defined by RFC 3986
 - o 0x04 : NFC Forum external type
 - o 0x05 : Unknown
 - o 0x06 : Unchanged (Used for a continuation chunk)

- 0x07 : Reserved

1.2.2. NFC Forum Smart Poster Record Type

NFC Forum has defined specifications for well-known data types based on NDEF format. Smart poster record type [16] defines the format which can be used to store URLs, text contents, mime data like images or similar data on NFC Forum compatible tags.

NFC Forum defines a Smart Poster use case [17] and this record type definition [16] is created for this use case. NFC Forum considers smart posters as one of the key use cases for NFC technology. An ordinary item like a poster on the wall can be turned into a “smart” item by attaching an NFC tag and the users can use their NFC enabled devices to read the data stored in the poster. NFC Forum [17] defines smart posters as:

Objects in or on which readable NFC tags have been placed. An NFC Smart Poster can come in many forms – it can be a poster, billboard, magazine page, even a three-dimensional object. The common factor is an NFC tag that has an NDEF message stored in it and is attached or embedded in the desired medium. This small tag with information is read when an NFC device is held close to it (p. 5).

Smart poster concept is basically built around URIs and optional actions. A smart poster must contain only one URI record. A URI can be used to define many kinds of data types and actions. URIs can be used for many purposes like “tel:1234567” to call a number, or “sms:1234567?body=url%20encoded%20sms%20content” to send an SMS.

Smart poster is actually an NDEF message which contains zero or more of the following record types in its payload [16] :

- Title record : A Text RTD instance which contains optional title of the poster
- URI record: there can be only one URI record in a smart poster, the URI record is considered as the core of the smart poster. Other records are all considered as supporting records for the URI.
- Action record: This record defines how data should be handled, but it is more like a suggestion the application may not follow this action. Action record type is a smart poster specific record type. Content is a single byte which

defines the action. Smart poster specification contains three predefined actions :

- 0: Do the action defined by the URI, for example send the SMS; dial the number etc.
 - 1: Save for later use, for example save the phone number in contacts.
 - 2: Open for editing, for example open the SMS editor prepopulated with data read from the smart poster.
 - Other values up to FF are reserved for future use (which we used to define social actions like “follow on twitter” in this project).
- Icon record: The smart poster may contain image records that will be used as the icon of the poster.
 - Size record: If the URI is referencing an external entity the poster may include the size of the external entity in this record, this record is optional.
 - Type record: If the URI references an external entity type record may be used to define the mime type of the external entity, this record is optional.
 - The smart poster NDEF message may also contain other optional application specific record types.

In practice a smart poster is an NDEF message which contains a single record which contains an embedded NDEF message as the payload which contains other NDEF records that contain the actual data.

When an NDEF message contains URI records at the top level in addition to the smart poster record, URI records are ignored and smart poster record is used [16].

Smart poster specification [16] provides sample smart posters. The Figure 2.2 is taken from smart poster specification. The red part marked on the following image shows the header for the main record. Record name value at offset 3 is “Sp” which is the constant used to define a smart poster data type. Offset 2 contains the length of the payload, which is an NDEF message in this case. The payload NDEF message may contain multiple NDEF records.

Offset	Content	Length	Explanation
0	0xD1	1	NDEF header. TNF = 0x01 (Well Known Type). SR=1, MB=1, ME=1
1	0x02	1	Record name length (2 bytes)
2	0x49	1	Length of the Smart Poster data (73 bytes)
3	“Sp”	2	The record name
5	0x81	1	NDEF header. TNF = 0x01, SR=0, MB=1, ME=0
6	0x01	1	Record name length (1 byte)
7	0x00, 0x00, 0x00, 0x0E	4	The length of the URI payload (14 bytes) (long format)
11	“U”	1	Record type: “U”
12	0x01	1	Abbreviation: “http://www.”
13	“nfc-forum.org”	13	The URI itself.
26	0x11	1	NDEF record header (SR=1, TNF=0x01)
27	0x03	1	The length of the record name
28	0x01	1	The length of the “act” payload.
29	“act”	3	Record type: “act”
32	0x00	1	Action = Launch browser

Figure 1.2 Sample smart poster definition taken from Smart Poster specification [16]

1.2.2.1. Smart Poster Use Cases by NFC Forum

The idea behind smart poster concept is basically built on the idea that users will access information stored on physical items like posters, billboards etc. using their NFC enabled devices. NFC tags containing data attached to the physical items can be easily scanned by users.

Smart poster specification contains the basic elements that can serve this purpose but smart poster concept can also be used to store/transfer more complex data types [16].

NFC Forum has released the logo (see Figure 2.3) for NFC enabled smart posters [7]. NFC tags should be marked with this logo so the users can be easily able to locate attached NFC tags. The tag should be placed under this logo so the user can scan the tag by simply touching his/her mobile device to the logo.



Figure 1.3 NFC Forum logo

NFC Forum has defined sample use cases for NFC in their Smart Poster white paper [17] and various marketing materials but those use cases, at least the use cases in the official white paper does not include social networking uses defined in this project.

One of the early examples of smart poster implementation is the smart posters used by Monaco tourist information system (see Figure 2.4) [17].



Copyright © WIMA 2010

Figure 1.4 A smart poster in action

Smart posters have played a key role in “Amazing Oulu” and “NFC future school” projects in Oulu Finland in 2008 [17]. Other similar test, proof-of-concept style projects have been carried out around the world but the technology has not yet reached real life, wide-spread use.

1.3. USES OF NFC

Uses of NFC are generally summarized as “sharing, pairing and transactions” [18]. Currently, the focus of major vendors and companies are concentrated on transactions, uses of NFC for payment systems. Google wallet is a very popular example of use of NFC for payments. In Turkey, Turkcell wallet (cep-t cüzdan) provides similar functionality.

NFC adoption for non-payment applications has been very limited until now. Vendors, including Google think of NFC as a payment technology. But in practice NFC technology can be used for all purposes contactless cards, QR codes etc. are used today [19]. After all NFC technology is a method of transmitting data between active-active or active-passive elements, so it can be potentially used in many ways. NFC Forum defines possible uses of NFC in daily life in the Figure 2.5 [4]:







Area	STATION AIRPORT	VEHICLE	OFFICE	STORE RESTAURANT	THEATER STADIUM	ANYWHERE
Usage of NFC Mobile Phone	 Pass gate Get information from smart poster Get information from information kiosk Pay bus/taxi fare	 Adjust seat position Open door Pay parking fee	 Enter/exit office Exchange business cards Log in to PC; Print using copier machine	 Pay by credit card Get loyalty point Get and use coupon Share information and coupon among users	 Pass entrance Get event information	 Download and personalize application Check usage history Download ticket Lock phone remotely
Service Industries	Mass Transport Advertising	Public Transport	Security	Banking Retail Credit Card	Entertainment	Any

Figure 1.5 NFC Uses cases defined by NFC Forum [7, 17]

As described in Figure 2.5, NFC can be used for any purpose that involves transfer of small amounts of data in close ranges. This can be adjusting your seat position, starting the engine of your car, paying fees, storing loyalty card information, performing social interactions etc. NFC enabled devices and NFC technology can simply replace any smart card, magnetic card, QR code applications. NFC and

mobile devices can allow us to consolidate multiple physical elements into a single device. For example, today most of us carry at least one credit card, several loyalty cards, pass cards which can all be stored in our mobile devices.

1.4. NFC SUPPORT IN MOBILEDEVICES

As of August 2012, NFC enabled mobile devices are not yet widespread [20]. Especially lack of NFC support in very popular Apple devices is a very important negative factor for widespread use of NFC.

Blackberry, Google, HTC, LG, Nokia and Samsung are among the major device vendors that have NFC enabled devices but number of models with NFC support are very limited, for example only 8 of the tens of device models of Nokia support NFC [20].

iPhone 5 was released on September 12, 2012 without NFC support indicating that Apple will not support this technology in the near future. Apple's decision will negatively affect the number of NFC enabled devices in use in the world.

1.4.1.1. Hardware Requirements

NFC support requires an antenna (see Figure 2.6) and the antenna should not be covered with a metal cover. For example if the antenna is placed in the backside of the device the back cover of the device must be plastic. As stated in [21],

Metal introduces difficulties for antennas in systems using radio communication in the far field as well as for antennas in inductively coupled systems working in the near field. For 13.56 MHz, 2-3 cm of air spacing between antenna and metal is sufficient to assure practically no negative effects from the surrounding metal. For NFC implementations in cell phones or laptops, 2–3 cm of air spacing is mostly not affordable (p. 25).



Figure 1.6 NFC antenna embedded in back cover of Huawei sonic, aka Turkcell T20.

It is possible to add NFC support to a mobile device using an external NFC device (see Figure 2.7). For example NFC enabled cases or antennas for mobile phones are available, so even if the device does not have built-in NFC support it is possible to add NFC support by using external devices. But this is not the best way for a smooth user experience.



Figure 1.7 An SD card with embedded NFC antenna.

Since NFC is still considered as a payment only technology most articles currently available state that a secure element is needed for NFC architecture. For example

Simon Burkard [22] has defined essential components of NFC as follows: *“For NFC in mobile devices essentially four components are required: A Host Controller, a NFC Controller, a NFC Antenna and a Secure Element”* (p. 4).

But this is not necessarily true for social network use case. Secure element is required for applications that require increased security like payment applications to store sensitive data like credit cards. But for social networking purposes such high security standards are not required. Blackberry platform already provides a “software card emulation mode” which allows the mobile device to act as a tag without using a secure element [23]. But this mode is not available on Android platform. Although it is technically possible, by default it is disabled and it can only be enabled by generating a custom android version. Device vendors or mobile network operators usually release their own version of Android platform when they are releasing an NFC enabled payment application. Accessing and using a secure element generally requires hardware specific customizations so the number of supported hardware devices is usually limited. For example, Google wallet runs on only Sprint Nexus S 4G devices.

CHAPTER II

SOCIAL NETWORKS

2.1. SOCIAL NETWORKS IN GENERAL

Social networking sites, especially Facebook and Twitter became world changing phenomenon in the last few years. They are used by billions of users every day, and a large majority of Internet users regularly use social networking sites [24]. People use social networks for sharing photos, connecting with friends or even organizing revolutions.

Today individuals use social networks for connecting and sharing, and businesses are more and more using social networks for publicity. Especially Twitter and Facebook are becoming the de facto advertisement and publicity channels [25]. Nowadays, many advertisements contain links to Facebook or twitter profiles instead of corporate web sites. Corporate web sites are losing their ground to twitter and Facebook pages. As stated in [26],

Social networks affect consumer behaviors. Social networks are organized around users and they form an interconnected social graph around users. Adjacent users in a social network tend to trust each other and tend to have common interests and users have a trust relationship (p. 1).

“Social networks are trusted because of shared experiences and the perception of shared values or shared needs” [27]. A customer's decision is strongly influenced by opinions of friends, acquaintances etc. For example, it is very important for brands to have more “likes”, “followers” or “shares” since they can reach people connected to the users' social graph and people are more likely to trust content shared by their “social network friends”.

In her book named “The Facebook Era” Clara Shih defines the importance of social networks for advertisement and businesses as follows:

The online social graph lets companies connect in a serendipitous, meaningful and positive way with their customers, prospects, stockholders, workers and potential employees. The online social graph offers an ideal business environment. Interaction costs are minimal. Creating relationships (such as with new customers) is a natural process [28].

For example, if your brand name becomes a Top Trending topic on Twitter that may be equal to an advertising budget of millions of dollars. In addition, when you have hundreds of thousands of followers it is even better than a marketing budget worth several millions of dollars. Many companies and celebrities have “social network consultants” now.

It is almost impossible to see a web site without “share on Facebook”, “like” and “follow on Twitter” links. These are now the de facto standards of website-user interaction. Facebook “Like” button went viral and was integrated to millions of sites in a short period of time, now it is almost impossible to find a site without a “like” button. Facebook likes and shares increases traffic for websites which in turn means increases in revenues. Twitter also provides the same traffic effect for websites. Social networking sites are very important factors for the entire web economy.

2.2. SOCIAL NETWORKS IN A MOBILE CONTEXT

“Pervasive” and “ubiquitous” are the keywords defining computing in this new era of computing centered on mobile devices. With the advancement of mobile devices, especially phones and tablets, ubiquitous computing became the reality. Mobile devices are always on, always with the user and they are easily accessible. And they are getting more and more powerful and capable every day.

Ben Dodson and Monica Lam define the role of mobile devices as follows in their paper named “P2P Micro-Interactions with NFC-Enabled Mobile Phones” [29]:

The smart phone, being powerful, personal, always with us, always-online, is changing our everyday life. It will eventually hold the key to our identities, access rights to digital assets, personal communications, photos, media, etc. In a sense, the smart phone is an extension of our digital self... (p. 1).

According to a comScore research in March 2012 (US smartphone subscribers over age 18 on iOS, Android and RIM platforms): 78 million users visited Facebook using mobile devices and on average they spent 441 minutes on Facebook using a mobile device. In the same period approximately 26 million users visited twitter using

mobile devices and they spent 114 minutes on average. The numbers are huge, and it is a business worth billions of dollars.

According to a research named “The Social Habit” by Edison Research [24] percentage of Americans who currently have a profile page at any social networking website is increasing steadily over years. As of February 2012 more than 56 percent of the US population over age 12 have social network profiles (see Figure 3.1).



Figure 2.1 %Who Currently Have a Personal Profile Page Any Social Networking Website

The same research by Edison Research [24] also shows that almost two thirds of social network users use the sites every day. And almost 22 percent of the entire population visits social networking sites several times every day.

“The Social Habit” [24] research also shows that people use social networks own more smart phones than average US population (see Figure 3.2).

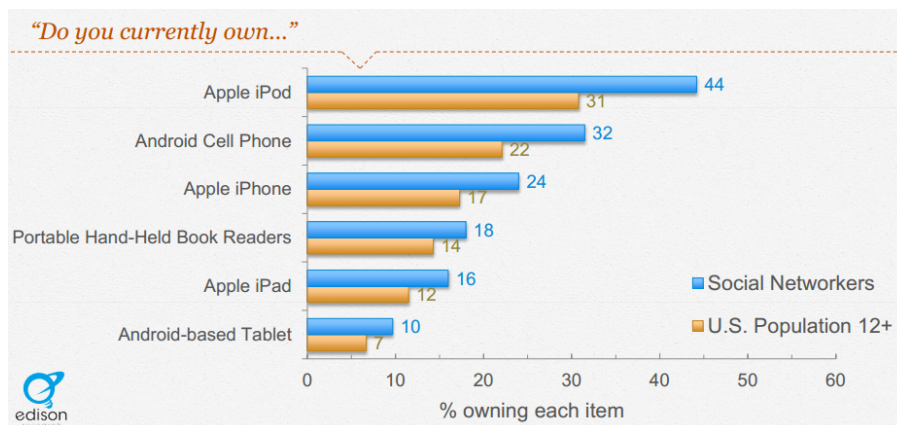


Figure 2.2 % of Social Network Users Owning a Smart Phone

2.3. FACEBOOK

As of August 2012, according to data published by Facebook, Facebook has over 900 million active users and more than half of them are using Facebook on mobile devices according to a research. 300 million photos are uploaded to the site daily.

“The Social Habit” report [24] by Edison Research support the same findings. According to Edison Research (see Figure 3.3), 54% of Facebook users use mobile devices to visit Facebook. The same research also shows that 33% of Facebook users use their mobile devices to access the site most.

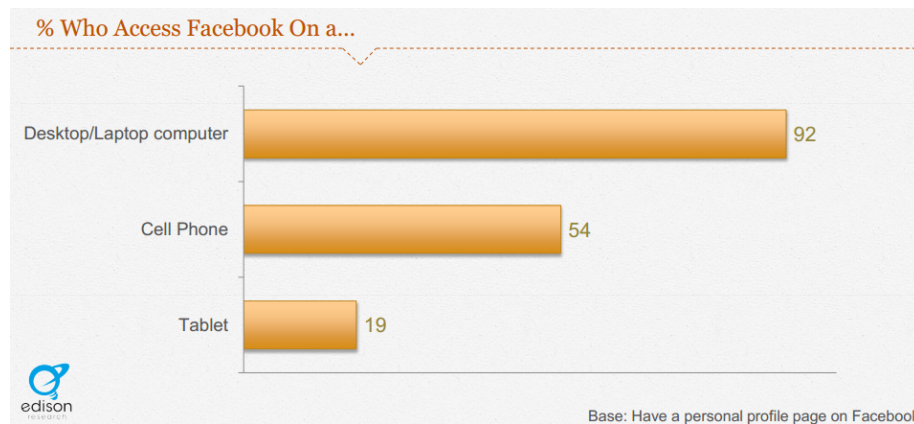


Figure 2.3 Distribution of Users by Device Used to Access Facebook

Facebook provides very powerful APIs for integration [30]. The real value behind Facebook is the data, information about its users. By providing APIs Facebook allows integration of its services into third party applications. So Facebook is not only limited to facebook.com domain, it is practically everywhere, it has codes integrated into millions of other domains.

Facebook provides applications and APIs for mobile platforms. Almost every user with a smart phone and a Facebook account has the “Facebook app” installed. According to Google Play Store statistics as of August 2012 Facebook Android application was downloaded more than 100 million times in 30 days [31].

2.3.1. Facebook NFC Use Cases

Main use cases of NFC for Facebook can be sharing and liking. In this study, because of the nature of NFC technology our main use case focus is the scenarios that involve physical interaction. Therefore, we will be focusing on scenarios in which users can physically interact with some items and share information stored in them.

2.3.1.1. Use Case 1 – Basic Usage:

Assume that you are an owner of a restaurant and you want your customers to like your restaurant on Facebook. Normally the users would have to navigate to a web page that includes the “Like button” that could be used to like your restaurant.

The main problem in this scenario is that the customers should spend too much effort to like your restaurant, the longer it takes less people will do it. Especially if you are a business like a restaurant that have physical interaction with their customers it may be harder to make the users visit a website and like or share content about your restaurant. To achieve higher success rates you should make the user complete the action while he/she is still in your restaurant. Only a very few people would bother to visit the website of the restaurant and like/share the restaurant after they return home. So the user must be able to perform the “social action” while he/she is still in the restaurant and very easily. The entire process should be only a few clicks/taps long, if it takes a more than 30 seconds or so the user will probably loose interest and would not bother to complete the process.

This entire process can be shortened dramatically using a simple NFC tag attached to the table or the menu. The NFC tag should contain all the necessary information to like/share on Facebook. When the user touches his/her NFC enabled device to the tag the device should read the tag contents and allow the user to perform social actions with only a few clicks/taps.

So by using NFC technology the entire process for the users can be reduced to a few simple clicks which would make more users complete the process.

Being able to like or share content in a very simple manner like this would increase the number of social interactions and increase popularity of the restaurant.

To accomplish this use case scenario the application is only required to read NFC tags and perform actions defined by the tag or navigate to a URL read from the tag. The rest of the process would be handled by standard Facebook functionality.

2.4. TWITTER

According to “The Social Habit” [24] research by Edison Research 10 percent of US population use twitter. The same research shows that one quarter of users that update their statuses on Twitter use mobile devices to update their statuses. And 61 percent of Twitter users own smart phones.

Twitter is a mass communication system, and allow users to broadcast their messages in a sense. Twitter has been defined as “an information company by its founders” [64] and it has been called as the “*electronic form of the word-of-mouth*” [32]. Word of mouth essentially represents “*the process of conveying information from person to person*” and it is becoming acknowledged as one of the most effective forms of marketing [32].

Tweets instantly reach followers and they propagate deeper in the social space through re-tweets. Twitter is now an important publicity and marketing platform [25]. Twitter practically replaced press releases; people now just tweet about of things they want to announce. Some politicians are actively using twitter to make announcements or declare their opinions on public matters.

One of the very first prerequisites of turning twitter into publicity or marketing platform for you or your business is to have more followers. If you don't have followers no one will know about your tweets, you will almost be like talking to yourself. Twitter followers are your audience, you can reach them any time and they can help spread your word by retweeting. Second key factor in a successful Twitter utilization for publicity is being able to control the tweet contents about your business. By adding hashtags or search engine optimized phrases it may be possible to generate more profit from Twitter.

2.4.1. Twitter Use Case

2.4.1.1. Twitter Use Case 1 – Follow

It is getting harder to find a corporate web site without a “Follow on twitter” link. First use case for Twitter is the “follow” scenario.

Assume that you are the owner of a restaurant and you want your customers to follow you on Twitter so that you can reach them later. If you can convince them to follow your restaurant while they are at the restaurant you will get the highest conversion rates (a conversion rate can be defined as ratio of users that follow the restaurant on twitter to the number of users that visit the restaurant), because most probably after they leave the restaurant, users will not go to your web site and follow you on Twitter. So you have to convince them to follow you on Twitter while they are still in the restaurant and they must be able to follow you with minimum effort.

By placing NFC tags on the tables or menus users can simply use their NFC enabled mobile device to read the tags and simply follow your restaurant with a few simple clicks.

2.4.1.2. Twitter Use Case 2 – Tweet

Assume that the restaurant owner in Twitter use case 1 wants users to tweet about his/her restaurant. But without using an enabling technology like NFC users would have to type the tweet manually, this may be considered as “too much effort” for most users and they would not tweet about the restaurant. So to increase number of tweets the process must be as simple as a few clicks/taps.

Even if the customers are determined enough to manually type and tweet, the restaurant will have no control on the tweet contents. If the restaurant can control the contents of the tweets they can generate more value from tweets. For example if they can control the tweet contents they can add hashtags, search engine optimization targeted content or some other content that they want to emphasize.

The restaurant can make the process much simpler and control tweet contents by placing simple NFC tags on the tables or menus (see Figure 3.4). The customer can simply use his/her NFC enabled device and tweet the tag content by a few clicks. Entire process will be much simpler and the business will have control over the tweet contents and generate more revenue.

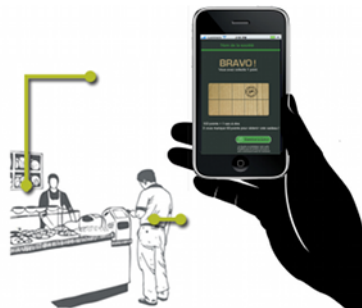


Figure 2.4 Customer physical interaction use case illustration

CHAPTER III

ANDROID PLATFORM AND NFC ON ANDROID

3.1. ANDROID PLATFORM

Android is a Linux based operating system for mobile devices like smart phones and tablets [33]. While primarily intended for mobile devices, Android is being used in all kinds of devices including home appliances, navigation systems, netbooks, smart TVs etc. Android was originally developed by Android Inc. and acquired by Google in 2005. Android is open source software with some exceptions.

Android became the leading smart phone operating system as of 2012, Android has 59% market share in smart phone market with more than 400 million active devices globally and 1 million new activations per day [31].

Top Smartphone Platforms 3 Month Avg. Ending Jul. 2012 vs. 3 Month Avg. Ending Apr. 2012 Total U.S. Smartphone Subscribers Ages 13+ Source: comScore Mobilens			
	Share (%) of Smartphone Subscribers		
	Apr-12	Jul-12	Point Change
<i>Total Smartphone Subscribers</i>	100.0%	100.0%	N/A
Google	50.8%	52.2%	1.4
Apple	31.4%	33.4%	2.0
RIM	11.6%	9.5%	-2.1
Microsoft	4.0%	3.6%	-0.4
Symbian	1.3%	0.8%	-0.5

Table 3.1 US smartphone platforms distribution by comScore research [3].

Primary development environment for Android is a customized version of the Java language [33]. Android contains a custom runtime environment named Dalvik [34] developed by Google. Some of the standard Java libraries can be used as is but

Android runtime environment is not totally compliant with standard java. Android does not support j2me, so it is not possible to run existing j2me applications on Android by default. It may be possible to run j2me applications on Android using some third party tools. But Android platform provides so many functionality and opportunities that developing native Android applications using Android APIs is the most effective option.

It is also possible to develop applications using the native development kit [35] and C/C++ languages but it is not as popular and as simple as Java platform. Java language and Android software development kit were used in this project.

3.1.1. Android Intent Mechanism

Android platform provides an intent [36] mechanism which can be used as an interprocess communication, messaging infrastructure. Android documentation [33] defines intent as “an abstract description of an operation to be performed”, and intents provide a late runtime binding mechanism between code in different applications. Android platform allows applications to register for intents and when an event occurs registered applications for the intent are invoked.

android.content.Intent [36] class is the core of the intent mechanism. When an event/intent is raised Android platform finds the appropriate application, service or broadcast receivers for the intent and invokes them, passing the intent data to the target application. It is possible to define intent filters so that the applications can filter out the events they want to receive.

Intents contain the following pieces of information [36]:

- Component name: The name of the target class. This value is optional. For example if component name for the intent is defined as “com.facebook.katana.LoginActivity” then the intent will be delivered to an instance of that class. If component name is undefined the intent resolution mechanism based on action, data and category information is used to find and invoke the best matching application. So it is possible to invoke any specific class by providing the package and class name information.
- Action: Action is a string denoting the name of the action. There are predefined actions defined by the Android platform classes, which are values like ACTION_CALL which means that the number passed in data field

should be dialed. Android classes contain several similar predefined actions. It is also possible to create application specific custom actions.

- Data: URI of the data to be acted on and mime type of the data.
- Category: Additional information about the type of component that should handle the intent.
- Extras: Additional data passed as key-value pairs.
- Flags: Flags provide additional information on how to launch an activity. Flag values are defined in `android.content.Intent` class.

Using the intent mechanism it may be possible to pass data to official Facebook or Twitter applications if they are already installed on the device. For example after reading the NFC tag an application can simply pass the data to the official Facebook application by raising intent with `ACTION_SEND` action. Intent mechanism and its usage are covered in more detail later in this thesis.

3.2. ANDROID NFC SUPPORT

NFC support for Android was introduced in API level 9, Android version 2.3. Android NFC API is in `android.nfc` package [37,38]. Android NFC API provides an abstraction for NFC hardware and provides necessary functionality to process NDEF messages. Android NFC API documentation is publicly available at the following URL: <http://developer.android.com/reference/android/nfc/package-summary.html> [38].

Android NFC API provides an abstraction for hardware and radio transmission related details and hides all those details from application developers. It is possible to access hardware or transmission related details but for applications that do not need extra security or hardware specific customizations generally it is unnecessary to dive into those details. An application developed using Android NFC APIs will work on all compatible Android devices. Like other abstractions provided by Android platform this abstraction greatly help development of NFC applications on Android platform. Without such an abstraction developers would have to deal with compatibility problems with tens of different hardware devices, tags, formats and technologies.

3.2.1. Tag Technologies As Used In Android API

NFC tags are based on several different technologies and have different features. Implementations of `android.nfc.tech.TagTechnology` [39] interface provide access to these technologies and technology specific features. `TagTechnology` implementations provide readers/getters to access properties of the tag or data and methods to perform I/O operations.

`Ndef` [14] implementation is also an implementation of the `TagTechnology` interface, which allows access to `NdefMessage` and `NdefRecord` objects. Android documentation suggest the use of Nfc Forum Type 1 to 4 tags for guaranteed compatibility, vendor specific `Ndef` formats may not be supported on all Android devices.

All NFC compatible Android devices must provide the following `TagTechnology` implementations [39, 40]:

- `NfcA` (ISO 14443-3A)
- `NfcB` (ISO 14443-3B)
- `NfcF` (JIS 6319-4)
- `NfcV` (ISO 15693)
- `IsoDep` (ISO 14443-4)
- `Ndef` on NFC Forum Type 1 to 4 compliant tags

`Ndef` format and technology is used in this project so other technologies supported by the Android platform are not mentioned in detail in this document.

3.2.2. `android.nfc.NfcAdapter`

`android.nfc.NfcAdapter` class is the core of the Android NFC API [38]. This class represents the NFC adapter of the device. `NfcAdapter` instance raises the necessary intents when tags are discovered. For example if an application has registered for `NfcAdapter.ACTION_NDEF_DISCOVERED` intent then the application will be called by the Android platform when an NFC tag containing an NDEF payload is discovered, and the data read from the discovered tag will be passed to the registered application.

Applications can check NFC support by using the `NfcAdapter`. `NfcAdapter` for the device can be accessed using `android.content.Context.getSystemService` method provided by Android platform. Android platform provides easy access to system

services like telephone, NFC, wireless networking etc using `getSystemService` method [33].

The following code snippet contains a sample code to check if this device has NFC support and if NFC is enabled.

```
/**
 * checks if device has NFC support and if NFC is enabled
 */
private void checkNfcSupport(){
    NfcManager manager =
        (NfcManager) getSystemService(Context.NFC_SERVICE);
    NfcAdapter adapter = manager.getDefaultAdapter();
    if (adapter == null){
        showMessageToUser("NFC hardware not found");
        return;
    }
    if(!adapter.isEnabled()){
        showMessageToUser("NFC support must be enabled first");
        return;
    }
}
```

3.2.2.1. Intents

`NfcAdapter` class defines three intents: `ACTION_NDEF_DISCOVERED`, `ACTION_TAG_DISCOVERED` and `ACTION_TECH_DISCOVERED` [36,33].

When a tag that contains an NDEF message is discovered `ACTION_NDEF_DISCOVERED` intent is raised by Android platform. The system inspects the first `NdefRecord` of the first `NdefMessage` object and looks for a URI, SmartPoster or MIME record. This processing allows applications to register intents for more specific tags. If the discovered tag contains Ndef data and there is at least one registered application for `ACTION_NDEF_DISCOVERED` intent then this intent

will be raised and ACTION_TAG_DISCOVERED and ACTION_TECH_DISCOVERED intents will not be raised.

Applications can also register for specific technologies on discovered tags using ACTION_TECH_DISCOVERED intent. This intent allows applications to define custom filters and capture tags with specific technologies. This intent is started after ACTION_NDEF_DISCOVERED and before ACTION_TAG_DISCOVERED. ACTION_TAG_DISCOVERED will not be started if any activities respond to this intent.

ACTION_TAG_DISCOVERED intent allows applications to register activities to capture tag data which do not match other intents.

When there are more than one registered activity for an intent Android platform displays the activity chooser dialog to the user. But activity chooser dialog would be very inconvenient for NFC applications since if the user moves the device away from the tag to choose an activity the tag will be lost and activity may fail. Since this problem may render the application unusable it is very important for NFC applications to define the most specific intent filter possible so that there is only one activity registered for the intent.

Every Android application must include a configuration file named AndroidManifest.xml file. Intents can be defined in this file which registers the activity for the specified intent. To capture NDEF_DISCOVERED events an intent filter similar to the following must be defined in AndroidManifest.xml:

```
<activity ...>
    <intent-filter>
        <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

Android platform will invoke this activity when an Ndef message is read from a tag. The data will be passed as an already parsed NdefMessage object so the application developer will be isolated from all transmission and parsing processes.

3.2.3. Permissions

Android platform uses application manifest files to define runtime permissions [41]. So permissions required by the application are known at install time and before the application is run.

Applications that want to use NFC features must declare this permission in their AndroidManifest.xml file. Name of the NFC permission is "android.permission.NFC".

NFC support must be enabled at device settings for the NFC hardware to generate or respond to NFC communication over its NFC radio. Even if NFC is enabled in device settings the device must be unlocked for NFC support to become active. If the device is locked then the device will not detect, read any tags or generate any NFC communication.

The following snippet shows a typical definition of NFC permission in AndroidManifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sozkan.socialtouch"
    android:versionCode="1"
    android:versionName="1.0" >
...
    <uses-permission android:name="android.permission.NFC" />
...
</manifest>
```

3.2.4. android.nfc.NdefMessage

android.nfc.NdefMessage [38] class is the class used to represent an NDEF message. An NDEF message is basically a container for one or more Nfc records. First NdefRecord in NdefMessage object is important since the intent dispatch mechanism of NfcAdapter class relies on the first record in a message, fine grained intent filtering (based on mime type, uri etc.) is based on the first record in the message.

NdefMessage class contains methods to create NdefMessage instances either by parsing raw binary data or from one or more NdefRecord objects. NdefMessage constructor that parses raw binary data performs necessary validations and saves the developer from manually validating the message format. But it is the application developer's responsibility to validate the data contained in the constructed object against the requirements defined by the NFC Forum [38]. For example the data may have the correct binary format but it may contain inconsistent data and NdefMessage constructor does not raise an error in this case.

Android by default does not provide classes to process NFC Forum smart posters, so necessary classes to process smart posters were developed for this project. But since smart posters are based on NDEF messages it is relatively simple to develop smart poster handling code. This is again one of the advantages of developing NFC applications on Android platform. The platform already provides almost everything needed to develop NFC applications. If the platform did not provide necessary NdefRecord and NdefMessage classes it would be harder and possibly error-prone to parse the data.

3.2.5. android.nfc.NdefRecord

android.nfc.NdefRecord represents an NDEF record. Ndef records contain typed data such as URIs, mime type data or application specific data.

NdefRecord class after Android API level 14 contains utility methods to create new properly formatted NdefRecords [14, 38]. For example createUri method can be used to create an NdefRecord object that contains a URI. But since the target platform of this project is API level 10 it was not possible to use those methods introduced in later versions and similar methods were developed for this project.

The following code snippet contains a method used to create a TextRecord [62].

```
/**
 * creates a text record. First bytes of a text record contains language
 * code length and language code
 * @param languageCode
 * @param text
 * @return
 * @throws IOException
 */
public static NdefRecord createTextRecord(String languageCode,
                                           String text)
throws IOException{
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    baos.write(languageCode.length());
    baos.write(languageCode.getBytes());
    baos.write(text.getBytes("UTF-8"));
    NdefRecord record = new NdefRecord(NdefRecord.TNF_WELL_KNOWN,
```

```
        NdefRecord.RTD_TEXT, EMPTY_BYTE_ARRAY, baos.toByteArray());  
        return record;  
    }  
}
```

3.2.6. android.nfc.Tag

android.nfc.Tag class [38] represents a raw NFC tag discovered. Tag represents the state of the tag at the time of discovery. A new Tag object is created every time a tag is discovered even if the same tag is rediscovered (for example when the user moves the device and the tag is removed and then the device is returned into range), and only the last Tag object can be used.

The most important method of Tag class is getTechnologyList method which returns the list of technologies supported by the tag. This information can be used to perform technology specific actions on the tag.

3.2.7. android.nfc.tech.Ndef

android.nfc.tech.Ndef class [38] provides the necessary methods to retrieve and modify the NdefMessage object on a tag. All Android device implementations are required to implement this class and support Ndef format, so Ndef format can be used for maximum device compatibility.

This class contains I/O methods and Ndef specific methods like getNdefMessage and writeNdefMessage. If using an ACTION_NDEF_DISCOVERED intent it is not necessary to directly use this class to read Ndef messages, this class should only be used for lower level processing of tags.

3.3. TAG DISPATCH

When NFC is enabled and the screen is unlocked, NFC enabled Android devices constantly scan for NFC tags[40]. When a tag is discovered the tag contents are parsed and the most suitable application registered for the tag type is invoked by the Android platform using a four stage dispatch mechanism [40] (see Figure 4.1).

1. Foreground activity dispatch: The application in the foreground of the device has called NfcAdapter.enableForegroundDispatch method, read NFC tags will be passed to this application. Other registered applications for the intent will not be invoked. This mechanism can be used to override default intent mechanism and capture all tags as long as the application is in the foreground of the device.

2. Ndef data dispatch (NfcAdapter.ACTION_NDEF_DISCOVERED intent)
3. Tag technology dispatch (NfcAdapter.ACTION_TECH_DISCOVERED intent)
4. Fallback to default tag discovered intent (NfcAdapter.ACTION_TAG_DISCOVERED intent)

Ndef format is used in this project for simplicity and maximum compatibility. Using a custom tag format or data format may cause compatibility problems or may add complexity to implementations. Considering that NFC Forum [7] is the leading organization and specifications published by NFC Forum has/will have the highest chances of adaptation, using Ndef messages can be considered as the best option.

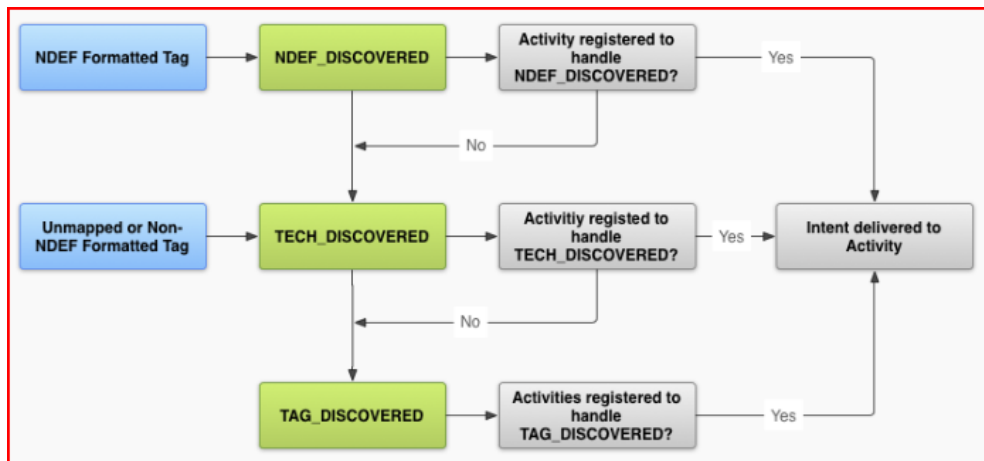


Figure 3.1 Ndef tag dispatch flow chart [33]

3.3.1. NFC Modes of Operation

Android NFC support is by default limited to Read/Write mode and Peer-to-Peer mode. Read/Write mode as expected allows devices to read and write tags. Peer to peer mode is introduced in newer versions and Android version 2.3.3, which is the target platform of this project, does not support peer-to-peer mode. Peer to peer mode allows two devices to communicate using NFC technology. “Android Beam” feature introduced in latest versions of Android uses this technology.

Card emulation mode is by default disabled in standard releases of Android. To enable card emulation mode the Android source code must be modified and a custom Android version must be created. (Android 4 and newer versions contain some related functionality but it is not yet real card emulation mode). This is mostly done to prevent possible poor user experience [33]. Since NFC on Android is usually considered in a payment context, card emulation mode is assumed to be used for payment applications. Card emulation mode when used for a payment

application requires access to a secure element, which is a sensitive and very restricted environment. Google has preferred to disable smart card emulation mode by default to prevent possible misuse of the APIs to access secure elements.

When NFC is used for non-payment applications card emulation mode actually does not require access to a secure element since sensitive data is not involved. For example, it may be possible to use card emulation mode to emulate a simple NFC tag which only contains some URL to share or some other non-sensitive information. Many companies, including Google, still think that main use case of NFC on mobile devices is wallet, payment or ticketing applications and they ignore other potential use cases.

RIM [23] recently announced software card emulation mode for Blackberry platform which allows applications to emulate simple NFC tags without using a secure element. Software card emulation mode may allow application developers to develop non-payment applications including social networking applications. For example if software card emulation mode was available on Android platform then it would be possible to develop an application that can be used to share simple social networking information with any other device, not only mobile devices, capable of reading NFC tags. But to do this using Android, with currently available features, both devices must be capable of peer-to-peer communication, which limits number of compatible devices and possible uses of NFC.

In my opinion, one of the most important problems that limit widespread use of NFC in mobile devices is the fact that major vendors, especially Google, consider NFC only in a payment or security related context. But when NFC becomes adapted for non-payment, non-sensitive activities, especially for social networking, it will reach widespread use. And for maximum compatibility with non-Android devices software card emulation mode, which would allow mobile devices to emulate simple NFC tags, is of paramount importance.

CHAPTER IV

PROPOSED NFC ENABLED SOCIAL NETWORKING SOLUTION

Ben Dodson and Monica S. Lam, in their paper titled “P2P Micro-Interactions with NFC-Enabled Mobile Phones” [29], has defined micro-interactions as follows:

Micro-interactions are categorized by their nearly instantaneous occurrence. They are invoked quickly and naturally, requiring no pre-configuration or setup. They occur with an absolute minimum amount of friction and require no training. ... (p. 1)

The solution proposed in this project can be thought as a form of micro-interaction which enables the users to perform social networking actions using NFC technology with minimal effort. The proposed solution aims to allow users to interact with social networks with minimum effort and additionally giving more control over the social interaction to the initiating party involved in the social interaction.

Customer-restaurant interaction defined in sample Facebook and Twitter restaurant use cases defined in previous sections of this document can be named as “business to consumer micro-interactions” as per the definition of micro-interactions [29].

Chen et al (2011, [42]) defined the relationship between cyber and physical spaces and how NFC technology can be used to make this relationship easier and more intuitive. Chen et al has concluded that touch driven interaction brings full user controllability, fine grained accuracy and high usability. The purpose of this project is also developing an easy and intuitive method of utilizing NFC technology for social networking actions.

4.1.1. Solution From Social Networking And Android Platform Points Of View

Social networking actions (social networking actions like “Tweet”, ”Share on Facebook”, ”Follow on Twitter” etc. are referred as “social networking actions”

throughout this thesis) on NFC enabled mobile devices can be handled in three ways :

1. Using official native applications for each social network: Almost every social network has an official native application installed on mobile devices. Facebook and twitter applications are almost always installed on devices if the user has a Facebook or twitter account. According to Google Play Store [31] statistics, as of August 2012, official Facebook application was installed more than 100 million times and official Twitter application was installed more than 50 million times in the previous 30 days. Thanks to the Android platform and intent mechanism [30, 36] official applications can be invoked and used quite easily, and using the features of these native applications like single sign-on can simplify the entire process.
2. Using URIs that point to a web page which contains the necessary social networking related functionality: In this case the role of the tags and tag handling application is greatly simplified. The tags contain a link to a web page for example on the official site of the tag owner and that page contains the necessary share, tweet, follow etc. links. The user simply navigates to this page using the web browser and then clicks the links/buttons on that page to perform social networking actions.
3. Using SDKs of the social networks: Both Facebook and Twitter provide SDKs for native Android application development [30, 43]. It may be possible to develop applications using these SDKs. Since developing native applications using SDKs would give more control over the process this may be a preferred way of handling social actions. But it will make things much more complicated, since custom development and maintenance would be required for every supported social network and whenever the SDKs are updated applications would have to be updated too. This may create more compatibility problems than the advantages of having more control over the application.

This project is focused on the first two methods. A proof of concept application, named SOcialTouch, was developed to demonstrate the developed solution. The application is a native Android application developed using Android SDK [33]. Further details of the development process and application can be found in the following sections.

4.1.2. Solution from NFC Point of View

To define a solution for widespread use of NFC technology for social networking the following solution is proposed:

- 1- The NFC solution that can be used for social networking should be built on NFC smart poster specification [16].
- 2- Predefined action values defined by the smart poster specification can be extended with actions that will be interpreted by the application as “Like on Facebook”, “Tweet”, “Follow on Twitter”, etc. (Smart poster specification [16] reserves values 3 to 0xFF for future use, new values can be defined in this range)
 - a. When the application reads the NFC smart poster the data should be processed according to the action value. For example if the action is “Tweet” a “Tweet” button will be displayed to the user so the user will be able to tweet with a few clicks.
 - b. If the action defined in the smart poster is not one of these social networking actions then the application will display data read from the smart poster and a button that will allow the user to visit the URL with a browser.
 - c. To avoid frequently updating NFC tags and to have more flexible options, the tag creator, the restaurant in our sample use cases, may prefer to create tags that contain a link to a page which contains all necessary social actions. In this case instead of performing the social action using native applications or APIs the user may perform the action after navigating to the URL using a web browser. For example, the tags contain a link to a page at the web site of the restaurant which contains all the necessary social actions like “Like on Facebook”, “Follow on Twitter”. In this scenario, the user simply navigates to the URI and visits the page and performs social actions using the web browser.

4.1.3. Application Flow

A social networking aware NFC enabled application flow is basically composed of the following major steps: Read the tag, display tag contents to the user, find the appropriate social action, and perform the action.

The flow can be further detailed as follows (see Figure 5.1 for a graphical representation):

- 1- Prerequisite: To read an NFC tag the device must be unlocked and NFC support must be enabled in device settings [40].
- 2- The application may be started or not before reading a tag. If it's not manually started it will be started when NFC tag is discovered if the application has an intent filter for NFC related ACTION_NDEF_DISCOVERED action [36,38].
- 3- An NFC tag is read by the device and passed to the application using intent mechanism
- 4- Tag data is parsed into a smart poster [16] object
- 5- If the tag is not a smart poster an error should be displayed to the user and no further action should be performed for this tag
- 6- Smart poster data is displayed to the user. Action record of the smart poster object should be inspected and if the tag contains a social networking action appropriate button for the action should be displayed on the user interface. No action before user approval should be performed.
- 7- If the action field of the smart poster contains a social action and if the user clicks the social action button (e.g. "Tweet" button) :
 - a. Check if the official application for this application is installed
 - i. If official application is installed invoke the intent and pass data to the official application
 - ii. If the official application is not installed perform the action by using functionality developed using the social network SDK or navigate to the URL read from the tag using the web browser
- 8- If the action is not a social action no social networking action button is displayed to the user. If the action is not a social networking action or if the user does not want to click the social networking action button :
 - a. User may choose to navigate to the URI read from the smart poster.
 - b. User may choose to do nothing and ignore the data

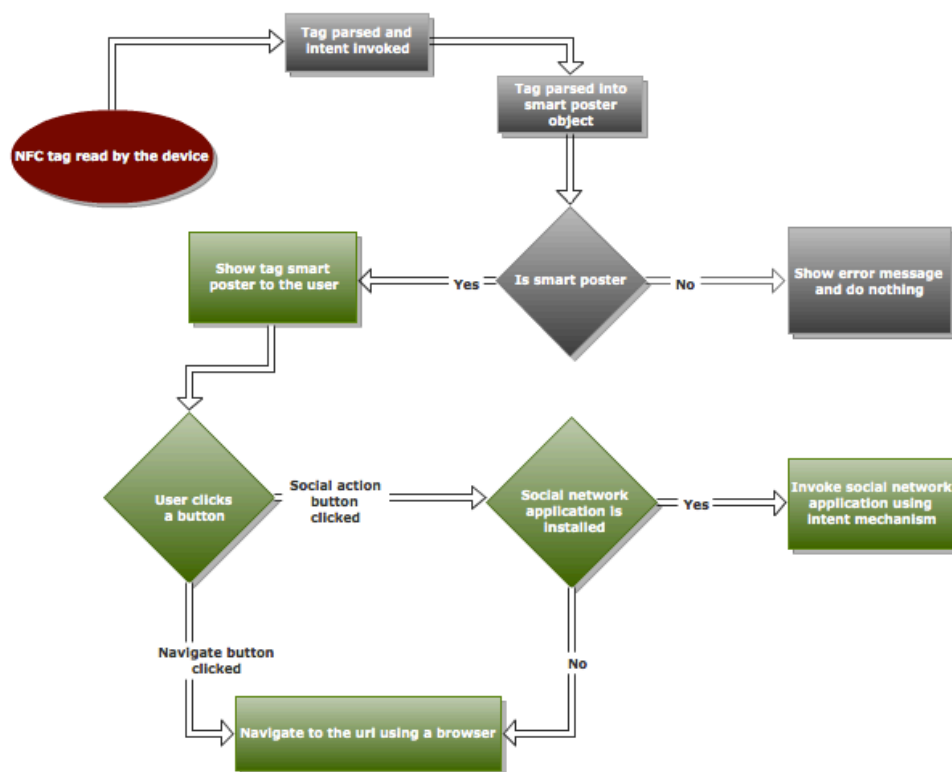


Figure 4.1 Flow diagram for the application

4.1.4. Security Concerns Related to the Solution

Smart poster use case is generally described as “the device reads the URI from the tag and navigates to the URI” [7]. If the application navigates to any URL read from NFC tags then the user may be target of potential attacks. The URL should be somehow validated before navigating. It may not be possible to validate the URL automatically, and user confirmation may be required. A URL validation service may be used before navigation but this may require some commercial subscriptions, services so this option will not be considered for this project.

So the application must request user confirmation or at least show the entire URL to the user before navigating to the URL. Otherwise the users have to trust the tag contents, which may not be always secure.

The tags must be write-protected to prevent unauthorized modification attacks. For example consider the restaurant case, if the tags are writable then an NFC enabled device and an application that can write to NFC tags can be used by malicious people to change contents of the tag. But once a tag is made read-only it is not possible to make it writable again. So it may not be possible to reuse NFC tags for

frequently changing content. So tag contents should be chosen carefully and content that would require frequent updates should be avoided. Actually by using an active device as an NFC tag, tag content can be dynamically updated but that would not be possible using passive tags.

There is also another potential security issue related to NFC tags, especially when they are used outdoors or unattended. They can be physically replaced with tags that contain malicious content. For example consider an NFC tag attached to an outdoor board without any physical protection, malicious people can physically replace the tag with a tag with malicious content.

It may even be possible to somehow break the original tag, for example by a physical effect, and replace a malicious tag behind or on top of it. For example after breaking the foreground tag sticker the attacker may place a tag on the other side of the poster.

To prevent these kinds of security problems applications must validate the URLs, as much as possible and require user confirmation before navigating. Directly navigating to a URL read from a tag may cause serious security issues for users. It is no different than clicking any link sent in an email, you may never know where you will end up.

So the developed solution does involve a manual approval process before blindly performing the actions read from the tag.

4.1.5. Future Use Cases

Since Android platform does not yet support software smart card emulation mode it is not practically possible to use Android devices as NFC tags. It may be possible to exchange social networking data between two Android devices using Android specific NDEF push protocol (com.android.npp [33]) or NFC Forum's Simple NDEF Exchange Protocol (SNEP) (SNEP is only available on devices running Android 4.0 and later [33]). But without software smart card emulation mode it is not possible to develop a solution compatible with most products. If it was possible to emulate simple NFC tags using the mobile device it would be possible to emulate tags dynamically, and create tag contents on the fly. So for example consider that you are the owner of a restaurant and you want to collect twitter accounts or email addresses of your customers. If the customer mobile devices were able to share that information by emulating an NFC tag it would be possible to increase amount of

successfully collected data. The users would simply share that information using their mobile devices. More social networking use cases can be defined in the future if it becomes easier to emulate a simple NFC tag using Android.

CHAPTER V

PREVIOUS RELATED WORK

5.1. WORK ON NFC AND SOCIAL NETWORKS

One of the most similar works to this study is the M.Sc. thesis completed in 2010 titled “Social Networks in an NFC Context” by Jonas Flodin of Lulea University of Technology [44]. The application was developed using j2me technology. Android 2.3 and NFC support was released on December 6, 2010; so when that thesis was completed NFC support for Android was not available. Jonas Flodin has also tried to demonstrate the possibilities of NFC in areas other than payment and ticketing but he had mainly tried to solve generic problems which are irrelevant using Android platform.

Android platform provides several features, APIs that allow application developers to interact with NFC tags and social networks [33]. Interacting with NFC tags is much easier than it was two years ago. But the technology is so new that there are not many researches, publications in this field and since it is changing almost every day existing work become out dated in a very short period of time. Some of the previous related works are listed below just to give an idea about what have been done in this area (this list is not an exhaustive list of related publications):

- “P2P Micro-Interactions with NFC-Enabled Mobile Phones” by Ben Dodson
Monica S. Lam [29].

This paper defines micro-interactions, simple actions that can be used to exchange information between devices using NFC technology. This paper is not directly related to social networking use of NFC technology but sets some interesting views which are applied in this project.

- “MyState: Using NFC to Share Social and Contextual Information in a Quick and Personalized Way” by Robert Hardy, Enrico Rukzio, Paul Holleis, Gregor Broll, Matthias Wagner [45].
A work from 2010 about using NFC technology to share information on facebook. The application was developed using j2me.
- “NFCSocial: social networking in mobility through IMS and NFC” [46]
A paper about an NFC aware social network application. The application uses SIP and IMS to transfer messages from the phone to the server. Developed using j2me.
- StoLPaN (Store logistics and payment with NFC) consortium was founded to overcome application level standardization and interoperability problems related to use of NFC technology for retail scenarios. StoLPaN consortium was founded by the members of the StoLPaN project consortium. StoLPaN project was run between 2006 and 2009 and was funded by the European Commission [47]. StoLPaN project was focused on payments, loyalty and other in-store retail scenarios but a social networking use case was not defined in this project.

A few years ago even Internet access or browsing web pages using mobile devices was a serious problem. Mobile technology is advancing at an incredible speed and papers or work older than 1 or 2 years are irrelevant, out dated at the moment, as this project will be in two years. For example using j2me is almost not even an option to develop NFC applications, since market share of j2me is diminishing and major platforms like iOS, windows phone and Android do not support j2me by default. Today almost everyone has either mobile internet access or many shops, restaurants etc. provide wireless internet access to customers, and mobile device browsers are almost as feature-rich and capable as desktop browsers [48]. Problems encountered two years ago are irrelevant at the moment.

5.2. WORK ON ALTERNATIVE TECHNOLOGIES

Vasquez-Brienso et. al. have studied Quick Response, QR, codes and NFC technology in mobile phones in their paper named “Using RFID/NFC and QR-Code in Mobile Phones to Link the Physical and the Digital World”[63].

QR code is a two dimensional barcode developed in 1994 by Denso Wave Corporation to improve reading speed of complex-structured two dimensional barcodes [63].

The main characteristic of a QR Code compared with a traditional bar code is that it contains information in both the vertical and horizontal directions, while a bar code contains data in one direction only. For this reason QR code holds a considerably greater volume of information. In addition it can encode several types of data including symbols, control codes, binary data, and multimedia data. The typical barcode holds a maximum of 20 digits, while the maximum data capacities of a QR code are 7,089 characters for numeric data, 4,296 characters for alphanumeric data, 2,953 bytes for binary data, and 1,817 characters for Japanese Kanji and Kana data. (p.222 [63])

QR codes can be scanned using mobile device cameras. QR codes can be used to transfer small amounts of data to mobile devices in a very similar fashion to transfer data to mobile devices using NFC technology. QR codes can be considered as an alternative to NFC technology.

	QR Code	NFC/RFID
Availability in mobile phones	High: Any camera-enabled mobile phone, several include preinstalled readers.	Low: Only NFC-enabled devices
Cost	Low: Tags can be printed in any printer, using common paper	Medium/High: Depends on the NFC/RFID tag or smartcard to be used.
Users Learning Curve	Low: Most users are already familiar with mobile cameras.	Medium: Users require learning NFC basis.
Security	Low: Information can be read easily by any camera-enabled device.	High: Devices must be very close to read information
Storage capacity	High	High
Damage resistance	Medium: QR Code includes error correction data that allows up to 30% recovery of a distorted or damaged tag.	Low: If wires are damaged tag cannot be read.
Visibility requirement	High: Code must be visible and well illuminated.	None: Tags can be hidden.

Table 5.1 QR code – NFC comparison table by Vasquez-Brienso et al [63]

Vasquez-Brienso et. al. have defined several use cases in which NFC and QR codes can be utilized to link physical objects to computing using mobile devices (see Table 6.1 for a comparison of NFC and QR Code technologies). But they have

not defined a use case for sharing data read from NFC tags or QR codes on social networking sites.

5.3. WORK ON SOCIAL NETWORKS

The term social network is a general concept which is not only applicable to social networking sites. Clemons, E.K. et al describes social networks as follows:

Social networks are familiar to all who study primates, from baboon troops and gorilla and chimpanzee groups to human societies at all levels of cultural development. A human social network can be a group of friends living within a city, or a group of college classmates who remain in frequent contact socially. It can also be a group formed specifically to accomplish a poorly structured set of tasks over time, like a business school MBA learning team.

(p.1-2 [27])

Obviously this study focuses on social networking web sites and refers to the term “social network” in this context.

The business use cases defined in this study are focused on using social networks for publicity and marketing. There are many works related to this topic but a few are included here for reference. One of the papers related to this area is “Customer Communications Management in the New Digital Era” (Hudson, S. et al 2012) in which the authors describe how marketers can leverage social media platforms to affect customer-business relations [49].

CHAPTER VI

APPLICATION DEVELOPMENT ENVIRONMENT

The application is developed using Android SDK [34] and Eclipse ADT [50] tools. The application is a native Android application.

6.1. SOLUTION ARCHITECTURE & REQUIREMENTS

An NFC enabled mobile device running Android version 2.3.3 or later with an active internet connection is required to run this application.

NFC hardware as described in the previous sections of this document must be available on the mobile device. The device must be unlocked and NFC must be enabled at device settings.

An active internet connection, either wireless or mobile network data plan must be available to be able to access social network web sites.

A Facebook or Twitter account is needed to be able to access those services. Facebook and/or Twitter official applications should be installed on devices for a smoother user experience.

6.2. ANDROID SDK

Android provides a software development kit which contains all the libraries and tools required to develop Android applications.

Each Android release has a different SDK version, called “platform” by Android. And there is an API level for each release.

Android documentation defines an API level as: “*API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform*” [51]. For example Android version 2.3.3 has API level 10. To develop applications compatible with Android release 2.3.3 the application must be developed and compiled using API level 10 SDK platforms.

Each new API level adds several new APIs and features but using features provided by latest API levels restricts number of supported devices.

Android version 2.3.3 is the most widely used version of Android as of August 2012. Google regularly publishes Android platform usage statistics at the following URL: <http://developer.android.com/about/dashboards/index.html> [52]

As of August 2012 Android version 2.3.3 is used by more than 60% of all Android devices worldwide (see Figure 7.1).

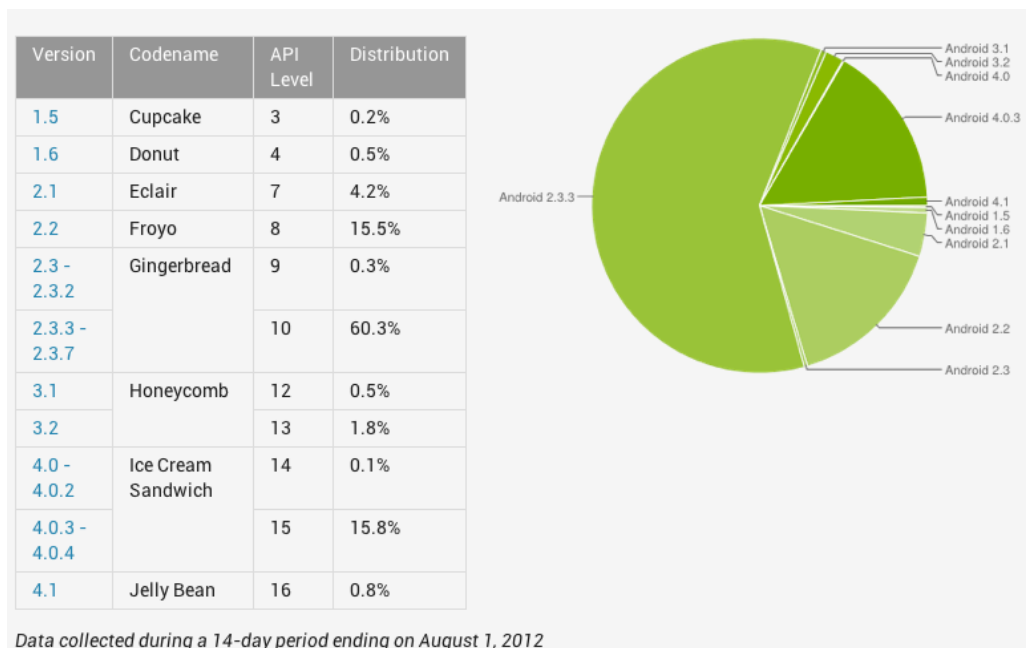


Figure 6.1 Distribution of Android devices by versions (Data collected from Google play store access statistics) [52].

Applications developed using an API level, for example 10, runs on all devices running newer versions of Android. For example, if an application is developed for API level 10 then it will run on Android version 2.3.3 and later versions. According to the data in the previous table, an application compatible with version 2.3.3 will run on almost 80% of all Android devices without any problems.

Android 2.3.3 is selected as the target platform in this project. So the developed application will be compatible with API level 10 and support almost %80 percent of existing devices (of course if they have NFC hardware support). Android 2.3.3 is also selected due to the fact that I own a 2.3.3 device (a Turkcell T20, a rebranded Huawei Sonic) with NFC hardware.

One of the main problems of Android platform is that there are so many versions of Android in use. Network operators or device manufacturers use different versions of

android platform. And most of the time it is not possible to upgrade the Android version to a later version due to restrictions imposed by the hardware, operator or manufacturer. So when you buy a brand new Android device it may be running an older version of Android and it may not be possible to upgrade it to the latest version. So it is very critical to support as many versions as possible. Many Android users will be using the Android version installed on the device when they bought the device through the entire life time of the device.

6.2.1. Android SDK Installation

Before Android SDK can be installed a Java SE development kit, JDK, [53] has to be installed first. Android SDK is dependent on the Java development kit, JDK. Android SDK can be downloaded freely from <http://developer.android.com/sdk/index.html> [34]. This initial download is not the complete package but it contains necessary tools for further setup. After installation of the initial download package “SDK Manager” application is run to perform actual SDK installation (see Figure 7.2).

SDK Manager allows us to choose which packages, API levels to install or uninstall. It is a straight-forward process to install Android SDK using the SDK manager.

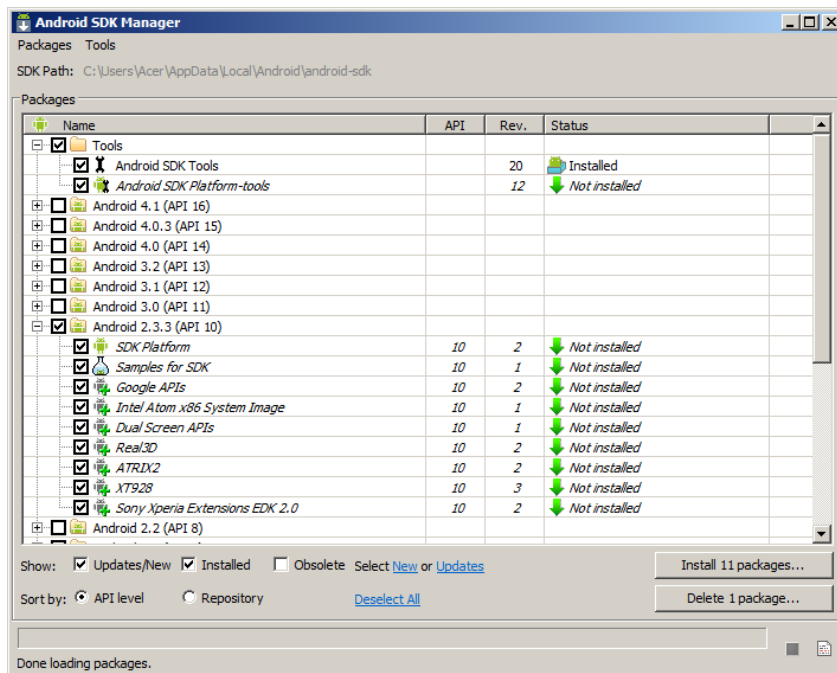


Figure 6.2 Android SDK manager GUI

6.3. DEVELOPMENT ENVIRONMENT

Eclipse [54] and ADT (Android Development Tools) plugin [50] was used for this project. Eclipse is an integrated development environment which is described as “a *multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system.*” [54]. Android development tool, ADT, is a plugin for Eclipse IDE and provides developers a powerful, integrated development and debugging environment to developers.

ADT can be installed using the documentation located at Android developer web site, <http://developer.android.com/sdk/installing/installing-adt.html> [50]. ADT is installed as an ordinary Eclipse plugin using “Install new software” menu and adding ADT update site to eclipse. To install the plugin Eclipse must be run as a user with administrator privileges.

Another way to install ADT is to install Google plugin for eclipse from <http://dl.google.com/eclipse/plugin/X.Y> (where X.Y stands for the eclipse version used). This eclipse update site contains both Google plugin for eclipse and ADT (besides other Google plugins not related to this project).

After installation the plugin must be configured and the Android SDK installation path must be configured, if not automatically configured by the installer. After installation of ADT creating a new Android application is quite straight forward using the Eclipse File->New menu.

6.3.1. Debugging

Android applications can be run or debugged using Eclipse and ADT [55]. The application can be either debugged on a virtual device or a hardware device [56] connected to the host computer via USB cable. Android SDK and ADT provide all the necessary tools to run and debug Android applications.

6.3.1.1. Virtual Device

Android SDK tools contain virtual devices that can be used to debug and test applications. A virtual device is defined as “*An Android Virtual Device (AVD) is an emulator configuration that lets you model an actual device by defining hardware and software options to be emulated by the Android Emulator*” by Android documentation [33].

“Android Virtual Device Manager” application should be used to create a virtual device, AVD, before first debugging attempt (see Figure 7.3). It is possible to create

a virtual device emulating any version of Android. An Android 2.3.3 virtual device was used to test this project.

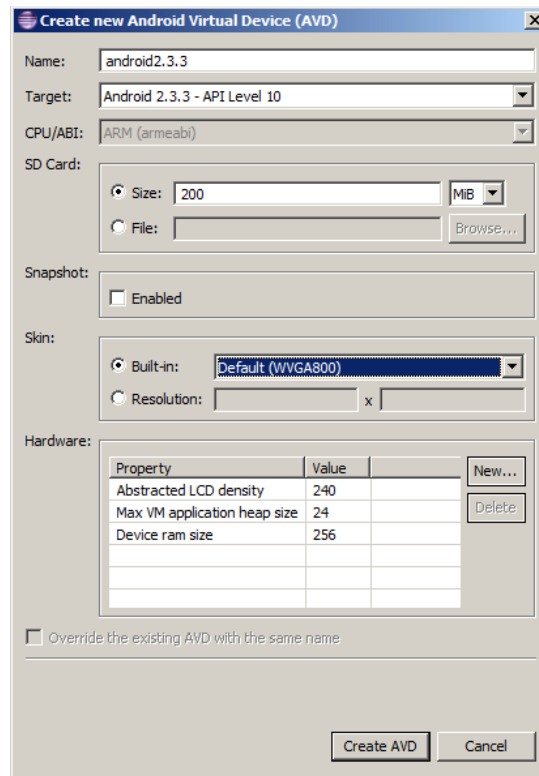


Figure 6.3 Android virtual device configuration

6.3.1.2. Debug Configuration

A debug configuration must be created before debugging the application for the first time [55]. Debug configuration allows the developer to choose the Android Virtual Device that will be used to launch the application.

Android requires that the applications to be digitally signed before running. When the application is run for the first time Android SDK tools create necessary debug keys and key store which will be used to sign the application. It is possible to use a self-signed certificate for development and testing but before releasing our application to production environment, like Google play store, the application must be signed using a valid certificate [55].

6.3.2. Testing

6.3.2.1. Unit Testing

To unit test applications Android Junit testing features provided by ADT and eclipse can be used [57]. The developer can create an Android test project linked to the

original project that contains the application, and can create test classes in the test project and run them. To develop an Android test case the developer should extend one of the Android test classes in `android.test` package [57].

After adding the class and writing test methods all the developer needs to do is right click the test case and click “Run as Android Junit Test”. The virtual device is automatically started and the test case is run by eclipse [57].

6.3.2.2. Functional Testing

Mock NFC tags defined manually as byte arrays or created using Android NFC APIs were used to test this project. Since it would not be feasible to use actual hardware devices and NFC tags to test the application, the application was developed and tested using mock tags created programmatically.

`com.sozkan.socialtouch.test.nfc.MockNfcTags` class was created to create mock NFC objects to mimic data read from NFC tags. Sample tags defined as byte arrays from Android NFC samples and complex smart posters created dynamically using Android APIs were used for testing purposes.

Utility methods to create NFC tags programmatically and dynamically were developed so it was possible to emulate and test NFC tags. Similar methods are available in Android API level 14 but they are not available in Android API level 10, which is the target runtime for this project. After getting familiar with Ndef record structures and of course with the help of Android APIs creating records are quite easy.

6.3.2.3. Applications Required For Testing On Virtual Device

To be able to send events to official Facebook or Twitter applications on test environment, virtual device, these applications have to be installed to the virtual device.

To install applications on virtual devices the following command can be used:

```
adb install full-path-to-apk-file
```

Facebook.apk file can be downloaded with Facebook sdk from Github [30]. Official Twitter application is not directly downloadable, it can only be installed via Google Play Store [31] but Google Play Store is not enabled on virtual devices. To install Twitter application on an emulator one should install the official Twitter application on an actual Android device and copy the apk file from the device to the host

computer on which the emulator is running. Then it is possible to install the apk on the emulator by using “adb install” command.

To learn more about the details of these applications the Android Apktool, which can be downloaded from <http://code.google.com/p/android-apktool/>, can be used to reverse engineer the apk files. Apktool generates the AndroidManifest.xml file so the list of activities defined by an application can be generated, this list then can be used to determine the class names and invoke a specific class of the application. For example it may be possible to find out the exact name of the class (which is actually `com.twitter.android.PostActivity`) to create a tweet using this method.

6.3.3. Testing Using Open-NFC

Open NFC project [58] provides necessary tools and emulators to test NFC features on virtual devices. Open-NFC Android Edition [58] contains a virtual device emulator with NFC support. Open NFC also provides another application named “Connection Center” which accepts connections from other machines and allow multiple virtual devices to communicate with each other and test NFC features. Using these applications two Android virtual machines can interact with each other simulating an NFC interaction.

This method was not used in this project since for the purposes of this project it was not actually necessary to emulate different tag technologies, since the focus of this project is limited with Ndef messages and smart posters at a higher level. Open-NFC can be used to test/develop more complex and lower level applications. So although it was considered and worked on Open Nfc project was not actually used for this project.

6.3.4. Testing On Hardware Devices

Eclipse ADT also allows debugging on hardware devices [56]. <http://developer.android.com/tools/device.html> contains necessary information to debug applications on hardware devices.

A hardware device, a mobile device or similar, can be connected to the development computer using a USB cable and the application can be deployed and debugged on the hardware device directly from eclipse. To be able to debug on the hardware device USB drivers, which are specific to the hardware used, must be installed to the development computer first [56].

CHAPTER VII

THE APPLICATION

The application is named `SOcialTouch`. Java package used is `com.sozkan.SocialTouch` (see Figure 8.1 for project details). Minimum targeted API level for this application is 10, which stands for Android version 2.3.3, Gingerbread. Application source code can be downloaded from the following URL for future reference:

<http://sozkan.com/ms-thesis/>

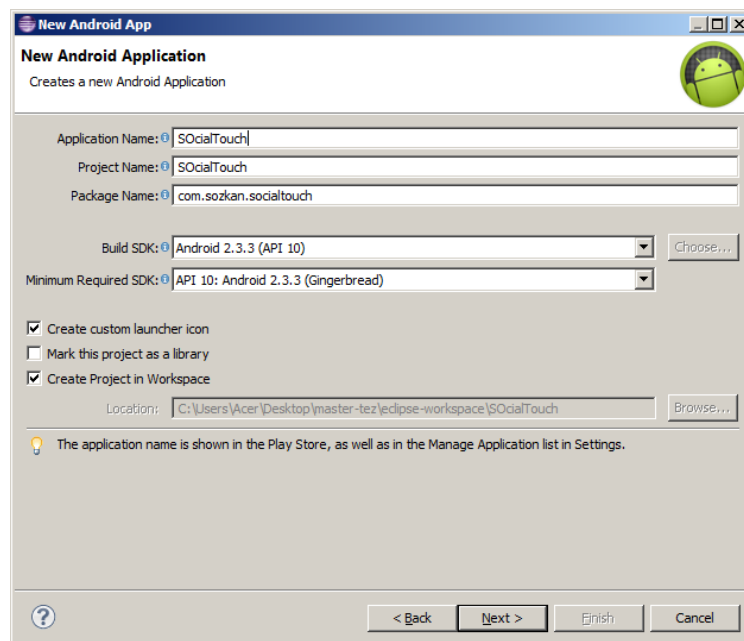


Figure 7.1 Application properties

7.1. PACKAGE STRUCTURE

- `com.sozkan.socialtouch` : Main package of the application
- `com.sozkan.socialtouch.nfc` : NFC related classes, including smart poster handling
- `com.sozkan.socialtouch.test` : Test classes, including mock tag and smart poster creation

For more information about package-class hierarchy see the class diagram of the application provided in Appendix A.

Use cases for the application can be seen in Appendix B.

7.2. APPLICATION CONFIGURATION

Every Android application contains a configuration file named exactly `AndroidManifest.xml` [59]. `AndroidManifest.xml` for `SOcialTouch` contains the required permission definitions.

The application requires internet access to function properly. To enable internet access the following line must be added to `AndroidManifest.xml`

```
<uses-permission android:name="android.permission.INTERNET"/>
```

To enable NFC access the following line must be added to `AndroidManifest.xml`

```
<uses-permission android:name="android.permission.NFC" />
```

The following `uses-feature` element allows the application to be listed at Google Play Store for only devices with nfc hardware.

```
<uses-feature android:name="android.hardware.nfc"
android:required="true" />
```

7.3. COMPONENTS USED

The application was built using standard Android classes and GUI elements. No third party library or components were used. `ViewFlipper`, `TableLayout`, `LinearLayout`, `Button`, `EditText`, `TextView` and `WebView`, provided by the Android platform, are the most significant components used in the project [33].

`WebView` [33] was used to embed a browser into the application. By embedding a `WebView` into the application it may be possible to navigate to the URIs read from the tags in a more controlled fashion. For example scripts may be disabled before navigating to the URI, which may provide some additional security for the user. Using an embedded `WebView` also by-passes the default intent mechanism and custom url scheme filters. For example if an embedded browser was not used a `ACTION_VIEW` intent would be used to open the URI but in that case some application registered for a specific url scheme might be used to open the URI.

7.4. IMPORTANT CLASSES AND METHODS

- `com.sozkan.socialtouch.SocialTouchActivity`: This class is the main activity of the application and contains all event handling and GUI codes. Important methods are listed below :
 - `checkNfcSupport`: Checks NFC support
 - `onResume`: This method is called by the Android platform when this activity is invoked, brought to foreground. If this method is called because of an `ACTION_NDEF_DISCOVERED` event, the parsed `NdefMessage` is obtained from the intent in this method.
 - `facebookShare`: Shares the URI read from the smart poster on Facebook
 - `twitterTweet`: Tweets the title of the smart poster
 - `twitterFollow`: Follows the username obtained from the title of the smart poster
 - `btnMockTagXClicked`: Methods used to generate mock tags and fill fields with these mock data
- `com.sozkan.socialtouch.NfcIntentGenerator`: Activity class used to generate a smart poster and invoke an intent with `ACTION_NDEF_DISCOVERED` action.
- `com.sozkan.socialtouch.nfc` package: This package contains NFC specific classes like smart poster and record handling classes
- `com.sozkan.socialtouch.nfc.SmartPoster`: NFC Forum smart poster implementation
- `com.sozkan.socialtouch.test.nfc.MockNfcTags`: Class used to generate mock NFC tags, smart posters. Contains utility methods used to create mime, text, uri records.

Please see Appendix A for the class diagram of the application.

7.5. USING OFFICIAL NATIVE APPLICATIONS FOR SOCIAL NETWORKS

SocialTouch application handles social actions by defining custom action values for smart poster action field in the unused 3-0xFF range [16]. When a smart poster is read, appropriate handling method is chosen using the action value. Custom action values are defined in `com.sozkan.socialtouch.nfc.SmartPosterAction` class. For example, tweet action is defined as seen in the following code snippet:

```
public static final byte ACTION_TWITTER_TWEET = (byte)20;
```

So when a tag which contains the value “20” (data type is bytes) in its action field is read the application will show a “Tweet” button to the user. And the title field is expected to contain the tweet text. If twitter application is installed on the device twitter application will be invoked and tweet text will be automatically passed to the twitter application so the user will not have to type the tweet contents.

Intent [36] mechanism is used to invoke other applications and pass data to them. For example the following code snippet shows invocation of official twitter application. Package name for the official twitter application is com.twitter.android and the class name of the activity used to post a tweet is com.twitter.android.PostActivity. In the following code snippet the official twitter application is invoked and title field of the smart poster is passed as tweet contents.

```
/**
 * opens twitter dialog with tweet text prepopulated with
 * smart poster title
 * the user has to manually click tweet button on that screen
 *
 */
private void twitterTweet(){
    Intent shareIntent = new
        Intent(android.content.Intent.ACTION_SEND);
    shareIntent.setType("text/plain");
    shareIntent.putExtra(android.content.Intent.EXTRA_TEXT,
        currentSmartPoster.getBestTitle(Locale.getDefault()));
    shareIntent.setClassName("com.twitter.android",
        "com.twitter.android.PostActivity");
    startActivity(shareIntent);
}
```

If twitter application is not installed the URI can be used as a fallback and the user can be navigated to the URI.

The following social actions and handling methods are defined in SocialTouch:

1. Facebook share: Facebook only allows sharing URLs using com.facebook.katana.ShareLinkActivity class of the official application. Only the URI field of the smart poster is used.

2. Tweet : `com.twitter.android.PostActivity` class allows tweeting. Title field of the smart poster is used as the tweet contents, other fields are ignored. It may be possible to share images by utilizing the icon field of the smart poster [16] but that use case is not implemented in this project.
3. Follow on twitter : `com.twitter.android.ProfileActivity` class displays the details of a Twitter account, this class is invoked to perform a follow action. The title field of the smart poster is expected to contain the twitter account name. URI field may contain a link to the twitter profile of the target user that can be used as a fallback mechanism if twitter application is not installed.

7.5.1. Custom URL Schemes

Custom URL schemes [60] are becoming more popular every day, especially on iOS and Android platforms. These platforms contain necessary mechanisms to easily define custom url schemes and URL handling associations. A custom url scheme can be easily defined at application level. For example, on Android platform, to register an application for a specific URL scheme all you have to do is define an intent filter in `AndroidManifest.xml` file of the application [59]. There is no need to make system level changes. This makes it defining and handling custom URL schemes much easier for application developers.

Many applications including official Facebook [61] and Twitter applications define custom URL schemes. So when standard actions like `ACTION_SEND` or `ACTION_VIEW` are performed with such URLs, official native Facebook or Twitter applications can handle them easily.

It is also a great opportunity for application developers because for example to invoke some functionality of official Twitter application they only need to invoke an intent with a URL having Twiter specific format, since official Twitter application will handle the url and the user can be redirected to a twitter page very easily.

For example the following URL opens the profile page of the user having username `theusername`.

```
twitter://user?screen_name=theusername
```

This mechanism can be used as a fallback mechanism for NFC smart poster applications. Even if the application that reads smart poster data is not capable of invoking a specific application when the uri in the smart poster is opened by the Android platform it will be passed to an application registered for the URL scheme.

For example assume that an application that processes smart posters encounters a uri that points to a social networking site that is not supported by the application. The application just needs to pass the uri to Android intent system [36]. Android platform will pass the data to the best application that can handle the uri.

But using URIs with custom schemes would limit the use of the tags. For example a URI with a `twitter://` scheme would be useless if an application that can handle `twitter://` scheme is not installed on the device.

SOcialTouch application does not pass URIs to Android intent mechanism for automatic handling because it aims to support only Facebook and Twitter and if the official applications are installed they are invoked using intent mechanism. Passing URIs to Android intent system for automatic handling can be only useful in cases where official applications are not installed but some other third party application capable of handling custom URL scheme is installed. In that case although official applications are not invoked a third party application would be able to handle the uri.

7.6. USING A COMMON WEB PAGE

The URI of a web page that contains necessary social networking actions can be used in the smart poster. So the user simply navigates to the page and performs whatever actions available on the page. This method can be used for more flexibility about tag and tag content management. When a change is required for example in case of a new twitter account name the tags would have to be updated which is a manual process. But if a common web page is used then modifying the page, without changing tags, would be enough. This method may have some disadvantages over using native applications like lack of single sign on or a poor user experience due to bad web page design etc.

When using this method the uri should be displayed to the user and navigation should take place after user approval to prevent potential security issues. Blindly navigating to any url may cause serious security issues in case of a malicious tag content situation.

7.7. USING SOCIAL NETWORK SDKS

Almost every social network provides an SDK/API [30, 43] which can be used to develop either native Android applications or web applications. Using SDKs to develop applications give much more control over the application than using intent mechanisms to invoke other applications to perform social networking actions. For

example Facebook application uses only the URL for sharing and ignores the text but using the Facebook SDK it may be possible to develop an application that will also automatically populate the text. But using SDKs has its limitations. Custom development would be required for every social network supported, and this would limit the number of supported social networks. Or when an SDK change affects the application it would be necessary to update the application which would in turn lead to more development and maintenance costs. Facebook SDK is shortly summarized, mainly to prove that it would require more effort than using other methods.

7.7.1. Facebook SDK

Facebook Android SDK can be obtained from the official Facebook developer site <http://developers.facebook.com/docs/reference/androidsdk/> [30].

Facebook Android SDK provides Facebook Platform support for Android applications. The SDK is developed and maintained by Facebook. The SDK enables single sign-on and access to Facebook Platform APIs like Graph API, and FQL.

To integrate with Facebook Platform the developer must have a Facebook account and the application must be registered with Facebook.

An id is generated for each registered application. Android application must be signed and Facebook requires the signing key hash so that they can check validity of the application. This validation process provides additional security for Facebook users but it also adds more complexity to the development and maintenance process [30].

So if an application is only Facebook specific it may be feasible to develop an application using Facebook SDK, but when a generic application that can handle many social networks is aimed using social network specific APIs or SDKs will not be feasible. This would require too much development and maintenance effort.

7.8. APPLICATION SCREENSHOTS

Since the application is developed for testing purposes it contains many elements just to be used for testing. Buttons at the top of the main screen are used for testing purposes and would not exist in a real application (see Figure 8.2).

Buttons labeled “Tag 1” to “Tag 4” are used to generate mock ndef messages to be able to test the application without actually generating/capturing ndef messages.

When “Tag 1” button is clicked and fields are filled with smart poster data generated on the fly by the application.

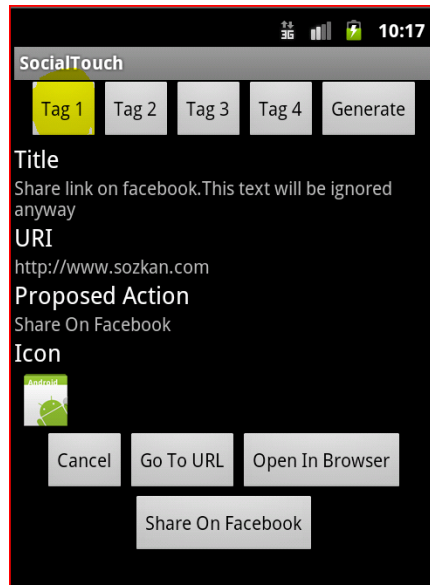


Figure 7.2 Application screenshot 1

Tag 1 button generates a smart poster with a “Share on Facebook” action, so a “Share on Facebook” button is displayed at the bottom of the screen. When the user clicks share on Facebook button native Facebook application is invoked this in turn opens a web browser and redirects the user to Facebook share link page. Official Facebook application automatically logs on the user. And the URL is automatically passed to Facebook application (see Figure 8.3). The user has to click on “Share” button manually to share the link but the process is very simplified.

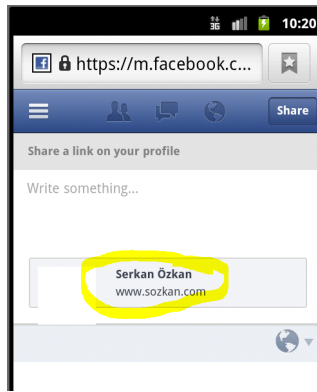


Figure 7.3 Share link on Facebook

Tag3 button generates a smart poster with a tweet action, and a “Tweet” button is displayed at the bottom of the screen (see Figure 8.4). When the “Tweet” button at the bottom is clicked the official twitter application is invoked and tweet dialog is opened.

The URI in this example uses a twitter specific URL scheme, so clicking “Open in Browser” button also opens the official twitter application since the uri has a twitter scheme that the Twitter application has registered. (Note: Some non-official applications also define filters for twitter URL scheme but official application is used in this example for the sake of simplicity)

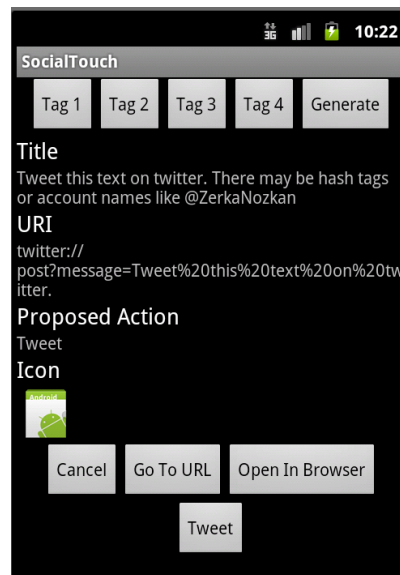


Figure 7.4 Tweet action

After clicking “Tweet” button the following dialog is displayed to the user (see Figure 8.5). The tweet text, title of the smart poster in this case, is passed correctly from our application to twitter application. The user only needs to click “Tweet” button.

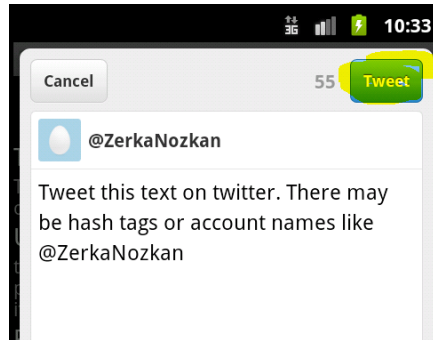


Figure 7.5 Tweeting

Regarding our restaurant scenario, the user is almost instantly ready to tweet the content created by the restaurant. The restaurant has control over the tweet contents and the process is as simple as a few clicks. Since we use the official twitter application the user does not need to logon to twitter if he/she had already signed in to the Twitter application.

“Tag 4” button generates a mock smart poster with a “Follow on Twitter” action. In this scenario, the tag contains a twitter username and our application allows the user to easily follow the user on twitter (see Figure 8.6).

Regarding our restaurant scenario the restaurant wants the customers to follow its twitter account. When the user uses his/her mobile device to scan the NFC tag attached to the table the user is directly taken to the twitter account and can follow the restaurant with a few clicks.

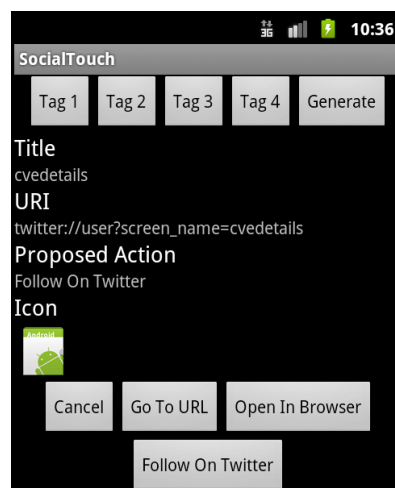


Figure 7.6 Follow on Twitter

When the user clicks “Follow on Twitter” button the user profile is displayed (see Figure 8.7) and the user only needs to click the “Follow” button to follow the user.



Figure 7.7 Following on Twitter

Generate button at the top right corner of the main screen opens another activity (see Figure 8.8) which can be used to generate custom smart posters and dispatch them to the main activity using the intent mechanism.

This activity can be used to generate smart poster data and test the application with arbitrary content. Smart posters generated by this activity only contain a URI, text and action for simplicity.

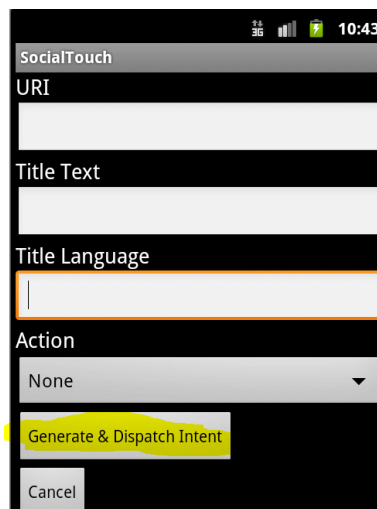


Figure 7.8 Dynamic tag generation

CONCLUSIONS AND DISCUSSIONS

In this study, we defined a social networking use case for NFC technology on mobile devices and developed a sample application which can be used as a baseline for fully featured, commercial grade applications. Establishing use cases other than payment applications for NFC technology is critical for the future and widespread use of the technology. It should be accepted that even if a technology is perfect in theory, it will eventually vanish if it cannot reach widespread use. NFC technology could have been considered as “new” in old times but in this rapid change era it’s getting older every day. NFC Forum [7] was founded in 2004 when there were no smart phones, no twitter, when Facebook was a small campus project etc. As of 2012, NFC technology possibly could not get as popular as expected back in 2004, but Android NFC support may greatly help the technology become popular as long as more devices with NFC hardware are manufactured and more realistic use cases can be defined for the technology.

Social networks are now one of the major forces shaping everyday life and the world. They are used for almost every purpose, from sharing personal issues to organizing revolutions. Mobile devices are spreading and advancing at incredible speeds. NFC technology and mobile devices can be used to improve social networking experiences. NFC technology may be used to simplify processes like sharing, following, advertising etc.

The following items form the conclusions of this study:

- NFC technology can be used to improve user experience and improve conversion rates in use cases that involve physical interaction and require simplified micro-interactions. For example, NFC technology may be a great enabling factor for the restaurant use case defined in this project. By using NFC technology users can perform social networking actions with a few clicks/taps and the restaurant may achieve higher conversion rates.

- Software smart card emulation mode support in Android platform would help establish new use cases for NFC technology by allowing Android devices to emulate simple NFC tags. This would allow Android devices to be used by simple sharing of information with NFC readers other than mobile devices.
- Smart poster specification [16] is aimed to be a general solution for many use cases; more specific specifications for social networking use cases would help development of social networking applications in a more standard way. Action values as they are defined for this project may be added to the specifications, so every developer would not need to define custom action values themselves, which would lead to chaos. In addition, instead of defining action values a more generic URL based but standard method can be developed to define social networking actions.
- Android intent mechanism can be used to invoke official applications of social networks. Using this method to perform social actions may simplify the process. Using a web page that contains all the necessary social networking actions can be used to simplify tag management procedures but since it would add another step (navigating to a page to tweet instead of simply clicking “Tweet” button) to the process it may adversely affect conversion rates.

RECOMMENDATIONS FOR FURTHER IMPROVEMENTS OF NFC TECHNOLOGY USE CASE FOR SOCIAL NETWORKING

First of all, major vendors like Google should focus on and push non-payment use cases of NFC technology. As long as their primary focus is on payment systems NFC technology will have less chance to achieve wide spread use. And obviously, more device manufacturers should add NFC support to their devices. But for that to happen first it should be proved that it is worth investing in NFC technology. As a rule of free market economy there should be more demand to create more supply, if there is not enough demand to convince the manufacturers to investing in NFC technology no one should expect the manufacturers to invest more in NFC technology. An Apple executive is said to say “It is not clear that NFC is the solution to any current problem” while explaining why Apple is not supporting NFC technology.

Social network oriented NFC specifications, whether built on smart poster specification or not, should be defined to prevent non-standards based, inconsistent implementations which usually cause major interoperability problems and prevent advancement of technologies.

RECOMMENDATIONS FOR FURTHER IMPROVEMENTS OF THE APPLICATION

First of all, the application developed as a part of this thesis is only a proof of concept application and it lacks a lot of features like ease of use, error handling etc. which are normally found in commercial grade applications.

Full smart poster handling support including images and multimedia sharing can be added to the application. Using these features the restaurant in the sample use case would be able to let its users share multimedia content about the restaurant.

Social network specific functionality using social network SDKs can be added to the application. This functionality may increase usability of the application for specific social networks but they can only be specific to selected social networks, and it may not be possible to develop a generic solution fit for all social networks.

A generic sharing solution like AddThis (www.addthis.com) can be used to support more social networks, platforms when using a common web page to perform social networking actions.

QR codes can be used as an alternative method of attaching information to physical objects. QR codes are more limited but for the purposes of this study and sample use cases QR codes also have adequate data storage capacity. QR codes can be scanned by any mobile device with a camera; unlike NFC support almost all mobile devices have built-in cameras. So QR codes can be used by almost all users, but QR codes have some limitations too. QR codes require the user to start QR code scanning application and take a picture of the QR code. This process involves more effort by the user than scanning NFC tags, scanning QR codes is more error prone, using NFC technology is more intuitive. But in any case QR code support must be added to the application to support devices without NFC support. Especially in the restaurant use cases QR codes can be used along with NFC tags.

REFERENCES

[1] Weiser, M. (1991) The computer for the twenty-first century, Scientific American, 265 (3), p. 94-104.

[2] Keller, K. (2009). Building strong brands in a modern marketing communications environment, Journal of Marketing Communications, 15(2/3), 139-155.

[3] comScore (2012), comScore Reports July 2012 U.S. Mobile Subscriber Market Share, http://www.comscore.com/Press_Events/Press_Releases/2012/9/comScore_Reports_July_2012_U.S._Mobile_Subscriber_Market_Share.

[4] NFC Forum, About NFC, <http://www.nfc-forum.org/aboutnfc/>

[5] ISO, ISO/IEC 18092:2004 Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1).

[6] Wikipedia, NFC article, http://en.wikipedia.org/wiki/Near_field_communication

[7] NFC Forum, Official Web Site, <http://www.nfc-forum.org>

[8] Google Inc., Google Wallet, <http://www.google.com/wallet/>

[9] Turkcell A.S., Turkcell Cep-t Cuzdan, <http://www.turkcell.com.tr/bireysel/servisler/Sayfalar/Turkcell-Cep-T-Cuzdan.aspx>

- [10] Sony Corporation, Sony Transferjet Technology,
http://www.sony.net/SonyInfo/technology/technology/theme/transfer_jet_01.html.
- [11] ECMA International (2004), Standard ECMA-340 Near Field Communication Interface and Protocol (NFCIP-1).
- [12] ISO/IEC (2005), ISO/IEC 21481:2005 Information technology -- Telecommunications and information exchange between systems -- Near Field Communication Interface and Protocol -2 (NFCIP-2).
- [13] ECMA International (2010), Standard ECMA-352 Near Field Communication Interface and Protocol -2 (NFCIP-2).
- [14] NFC Forum (2006), NFC Data Exchange Format (NDEF) Technical Specification.
- [15] NFC Forum (2006), NFC Record Type Definition (RTD) Technical Specification.
- [16] NFC Forum (2006), Smart Poster Record Type Definition Technical Specification.
- [17] NFC Forum (2011), White Paper: Smart Posters How to use NFC tags and readers to create interactive experiences that benefit both consumers and businesses.
- [18] Nosowitz, D. (2012), Everything You Need to Know About Near Field Communication, Popular Science Magazine.
- [19] Holmstad , Ø., Kreutzer, T. (2011), NFC Prototype Bonanza, Department of Computer Science Faculty of Science and Technology University of Tromsø, Norway.
- [20] NFCWorld.com (2012), A definitive list of NFC phones,
<http://www.nfcworld.com/nfc-phones-list/>

- [21] Rolf,E. Nilsson,V. (2006), Near Field Communication (NFC) for Mobile Phones, Department of Electrosience, Lund University.
- [22] Burkard, S. (2012), Near Field Communication in Smartphones, Dep. of Telecommunication Systems, Service-centric Networking Berlin Institute of Technology, Germany.
- [23] Research in Motion Limited, Blackberry Development Guide, Near Field Communication.
- [24] Edison Research (2012), The Social Habit, <http://socialhabit.com/>
- [25] Bulearca, M. Bulearca,S. (2010), Twitter: a Viable Marketing Tool for SMEs?, Global Business and Management Research: An International Journal Vol. 2, No. 4, 2010 pp. 296-309.
- [26] Mislove,A. et al (2007),Measurement and Analysis of Online Social Networks.
- [27] Clemons,E.K. et al (2007), The Future of Advertising and the Value of Social Network Websites: Some Preliminary Examinations, Wharton University of Pennsylvania.
- [28] Shih, C. (2009), The Facebook Era, Prentice Hall Direct.
- [29] Dodson, B., Lam, M.S. (2011), P2P Micro-Interactions with NFC-Enabled Mobile Phones, Computer Science Department, Stanford University.
- [30] Facebook Inc., Facebook SDKs , <https://developers.facebook.com/docs/sdks/>
- [31] Google Inc., Google Play Store, <http://play.google.com>

[32] Jansen, B.J. et al (2009) Twitter Power: Tweets as Electronic Word of Mouth, Journal Of The American Society For Information Science And Technology, 60(11):2169–2188.

[33] Google Inc. and The Open Handset Alliance, Android Developer Resources, <http://developer.android.com/>

[34] Android Documentation, Android SDK, <http://developer.android.com/sdk/index.html>

[35] Android Documentation, Android NDK, <http://developer.android.com/tools/sdk/ndk/index.html>

[36] Android Documentation, android.content.Intent Class, <http://developer.android.com/reference/android/content/Intent.html>

[37] Pelly, N., Hamilton, J. (2011), Google I/O 2011, How to NFC, <http://www.google.com/events/io/2011/sessions/how-to-nfc.html>

[38] Android Documentation, android.nfc API, <http://developer.android.com/reference/android/nfc/package-summary.html>

[39] Android Documentation, android.nfc.tech Package, <http://developer.android.com/reference/android/nfc/tech/package-summary.html>

[40] Android Documentation, NFC Basics, <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html>

[41] Shin,W. et al (2009), Towards Formal Analysis of the Permission-based Security Model for Android, ICWMC '09 Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications, 87-92.

[42] Chen, L. et al (2011), Touch-driven Interaction Between Physical Space and Cyberspace with NFC, Department of Computer Science, Zhejiang University Hangzhou, China.

[43] Twitter Inc. and other contributors, Twitter Libraries
<https://dev.twitter.com/docs/twitter-libraries>

[44] Flodin, J. (2010), Social Networks in an NFC Context, Luleå University of Technology Sweden.

[45] Hardy,R. et al (2010), MyState: Using NFC to Share Social and Contextual Information in a Quick and Personalized Way, UbiComp 2010, Copenhagen, Denmark.

[46] Fressancourt, A. et al (2009), NFCSocial: social networking in mobility through IMS and NFC, 2009 First International Workshop on Near Field Communication, Atos Worldline, Seclin, France.

[47] Medaglia,C.M. (2011), Chapter 15 Use Cases and Future Challenges for Near Field Communication: the StoLPaN Project, C.M. Deploying RFID - Challenges, Solutions, and Open Issues.

[48] Quirksmode.org, Mobile browsers,
<http://www.quirksmode.org/mobile/browsers.html>

[49] Hudson,S. (2012), Customer Communications Management in the New Digital Era, Darla Moore School of Business University of South Carolina.

[50] Android Developer Resources, Eclipse ADT Plugin,
<http://developer.android.com/tools/sdk/eclipse-adt.html>

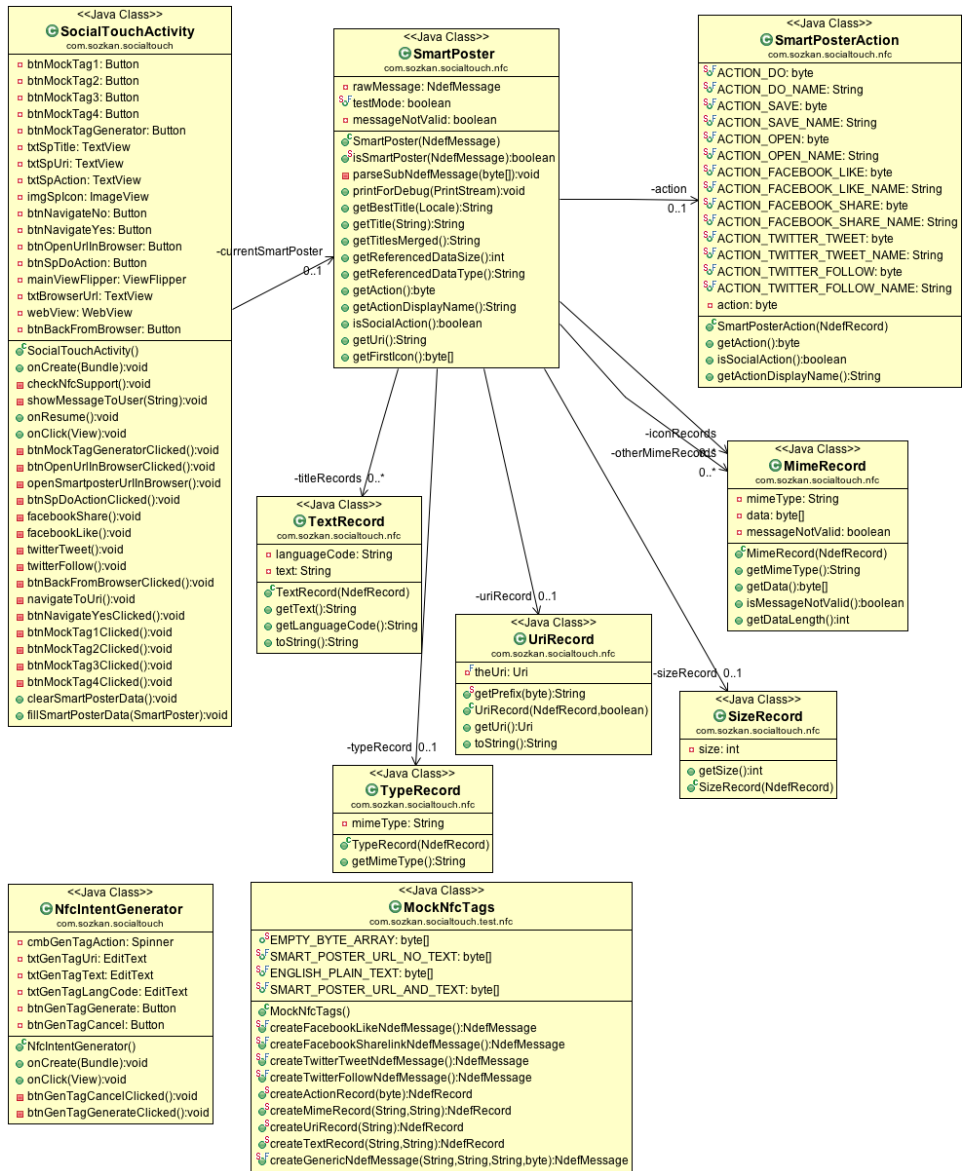
[51] Android Documentation, Android API Levels,
<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>

- [52] Google Inc., Android Statistics Dashboards,
<http://developer.android.com/about/dashboards/index.html>
- [53] Oracle Corporation, Java SE Overview,
<http://www.oracle.com/technetwork/java/javase/overview/index.html>
- [54] Eclipse Foundation, Eclipse IDE, <http://www.eclipse.org>
- [55] Android Documentation, Debugging from Eclipse with ADT,
<http://developer.android.com/tools/debugging/debugging-projects.html>
- [56] Android Documentation, Using Hardware Devices,
<http://developer.android.com/tools/device.html>
- [57] Android Documentation, Testing from Eclipse with ADT,
http://developer.android.com/tools/testing/testing_eclipse.html
- [58] Open NFC project, <http://open-nfc.org>
- [59] Android Documentation, The AndroidManifest.xml File,
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [60] Lee, T.B. et al (1998), RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax.
- [61] Constine, J. (2011), Facebook Pages Can Now Be Opened in Facebook's iOS Apps Via fb://page URL Scheme Links,
<http://www.insidefacebook.com/2011/10/21/facebook-url-scheme-pages/>
- [62] NFC Forum (2006), NFC Text Record Type Definition (RTD Text 1.0) Technical Specification.
- [63] Vazquez-Briseno, M. et al (2011), Chapter 12 Using RFID/NFC and QR-Code in Mobile Phones to Link the Physical and the Digital World, Interactive Multimedia.

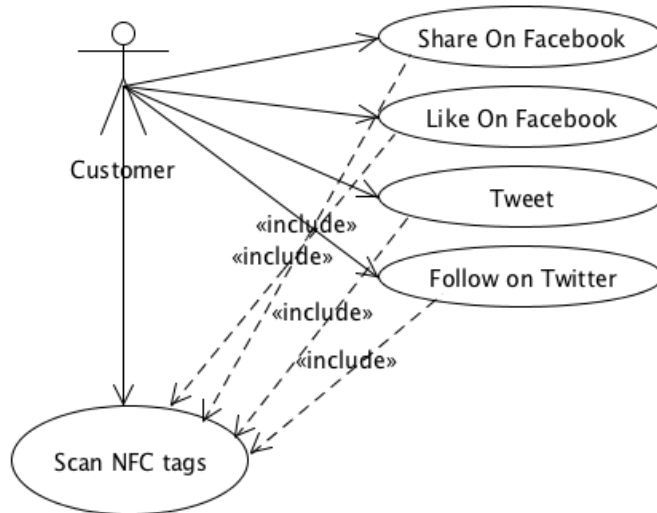
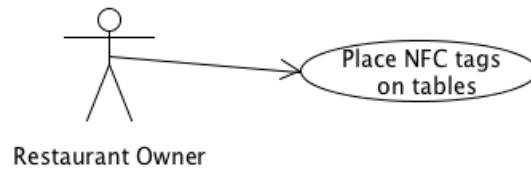
[64] The Economist (2010), Twitter's Transmitters,
<http://www.economist.com/node/15350950>

APPENDICES

Appendix A – SocialTouch Class Diagram



Appendix B – SocialTouch Use Cases



Appendix C – CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ÖZKAN, Serkan

Nationality: Turkish (TC)

Date and Place of Birth: 28 November 1976, Muğla

Marital Status: Married

Phone: 0533 632 30 11

email: serkanozkan@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
BS	Middle East Technical University, Civil Engineering	1999
High School	Bursa Fen Lisesi	1994

WORK EXPERIENCE

Year	Place	Enrollment
2012- Present	Ping Identity	Solution Support Engineer
2004-2012	Biznet	Consultant, Software Developer

FOREIGN LANGUAGES

Advanced English

HOBBIES

Word processing applications