

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gašper Urh

**Uporaba tehnologije NFC za potrebe  
mobilnih aplikacij na platformi  
Android**

DIPLOMSKO DELO  
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 00127/2013

Datum: 10.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GAŠPER URH**

Naslov: **UPORABA TEHNOLOGIJE NFC ZA POTREBE MOBILNIH APLIKACIJ  
NA PLATFORMI ANDROID**  
**THE USE OF NFC TECHNOLOGY IN MOBILE APPLICATIONS ON THE  
ANDROID PLATFORM**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

V okviru dela raziščite in predstavite delovanje tehnologije »Near field communication« (NFC) na mobilnih napravah, ki uporabljajo operacijski sistem Android. Uporabo tehnologije NFC demonstrirajte z izdelavo mobilne aplikacije, ki bo izkoriščala tehnologijo NFC. Poleg mobilne aplikacije izdelajte tudi strežniški del, ki bo skrbel zlasti za dostavo vseh potrebnih podatkov mobilni aplikaciji.

Mentor:

doc. dr. Damjan Vavpotič



Dekan:

prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Gašper Urh, z vpisno številko **63100300**, sem avtor diplomskega dela z naslovom:

*Uporaba tehnologije NFC za potrebe mobilnih aplikacij na platformi Android*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 11. septembra 2013

Podpis avtorja:



# Kazalo

**Povzetek**

**Abstract**

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Near field communication</b>	<b>3</b>
2.1	Razvoj . . . . .	3
2.2	Delovanje . . . . .	5
2.3	Varnost in vrste možnih napadov . . . . .	8
2.4	Razširjenost danes in možnosti za prihodnost . . . . .	10
<b>3</b>	<b>Tehnologije in oprema uporabljena pri razvoju aplikacije</b>	<b>15</b>
3.1	Android . . . . .	15
3.2	Java . . . . .	18
3.3	Android Studio . . . . .	20
3.4	Testno okolje . . . . .	20
3.5	SQLite . . . . .	20
3.6	Storitve REST . . . . .	21
3.7	JSON . . . . .	21
3.8	PHP . . . . .	22
3.9	MySQL . . . . .	23
<b>4</b>	<b>Načrtovanje in implementacija mobilne aplikacije</b>	<b>25</b>
4.1	Načrtovanje celotnega sistema . . . . .	26

## KAZALO

4.2	Implementacija strežniškega dela . . . . .	32
4.3	Implementacija aplikacije . . . . .	34
4.4	Analiza delovanja aplikacije . . . . .	48
<b>5</b>	<b>Zaključek</b>	<b>51</b>
	<b>Literatura</b>	<b>52</b>

# Seznam uporabljenih kratic

**NFC** - Near Field Communication  
**RFID** - Radio Frequency Identification  
**UPC** - Universal Product Code  
**EPC** - Electronic Product Code  
**API** - Application Programming Interface  
**SDK** - Software Development Kit  
**ADT** - Android Developer Tools  
**SQL** - Structured Query Language  
**HTTP** - Hypertext Transfer Protocol  
**REST** - Representational state transfer  
**SOAP** - Simple Object Access Protocol  
**W3C** - The World Wide Web Consortium  
**JSON** - JavaScript Object Notation  
**XML** - eXtensible Markup Language (XML)  
**WSDL** - Web Services Description Language  
**NDEF** - NFC Data Exchange Format  
**URI** - Uniform Resource Identifier  
**PIN** - Personal Identification Number  
**POS** - Point Of Sale  
**QR** - Quick Response  
**PDO** - PHP Data Objects



# Povzetek

Diplomsko delo na kratko predstavlja tehnologijo Near Field Communication, krajše NFC, skozi razvoj in delovanje. Obravnavana so razna varnostna vprašanja, trenutno najbolj razširjeni informacijski sistemi za mobilno plačevanje in tudi ostala področja uporabe. Kot primer uporabe tehnologije NFC je v delu predstavljen razvoj mobilne aplikacije NFC Traveler. Aplikacija je razvita v operacijskem sistemu Android, opisan pa je tudi podporni strežniški del. Predstavljene so vse potrebne tehnologije med razvojem in načrtovanjem na čelu z operacijskim sistemom Android. Omenjena aplikacija demonstrira uporabo tehnologije NFC na primeru turističnega vodiča. Z značkami NFC bodo označene posamezne turistične točke, katere bodo uporabniki lahko s pomočjo aplikacije obiskali in si ogledali njene podrobnosti. Omogočeno bo tudi pregledovanje vseh turističnih točk in podrobnosti brez predhodnega obiska.

**Ključne besede:** NFC, mobilno plačevanje, Android, aplikacija.



# Abstract

This thesis represents Near Field Communication technology, shortly NFC, via its development and functionalities. It also discusses various security issues, most widely used information systems for mobile payments, as well as other cases of use. One of them is developed as part of this thesis, mobile application for Android operating system called NFC Traveler. It is described with its supportive server side. Presented are all necessary technologies used for developing and planning of NFC Traveler, at the forefront of the Android operating system. Application will demonstrate the use of NFC technology and will be used as a tour guide. NFC tags will mark certain tourist spots, which users can visit with help of application and then view its details. It will also enable viewing of all the tourist spots and details without prior visit.

**Keywords:** NFC, mobile payments, Android, application.



# Poglavje 1

## Uvod

V današnjem času stremimo k avtomatizaciji vsakodnevnih opravil. V ta namen se razvijajo nove tehnologije in ena od teh je brezstična tehnologija Near field communication (NFC). Trenutno se uporablja predvsem kot podporna tehnologija aplikacij mobilnih naprav. Mobilno napravo lahko uporabimo namesto ključev stanovanja ali avtomobila, za izmenjevanje podatkov z drugo napravo, preverjanje prisotnosti na določenih lokacijah, itd. V tem delu je predstavljena konkretna uporaba tehnologije NFC na primeru interaktivnega turističnega vodiča za mobilne naprave.

Iz poslovnega vidika je področje mobilnega plačevanja tisto, ki naj bi v prihodnjih letih povzročilo hitro širitev uporabe tehnologije NFC. V tem trenutku že obstajajo različni plačilni informacijski sistemi, ki nadomeščajo oziroma nadgrajujejo trenutne plačilne metode s plačilnimi karticami in s tem pohitrijo plačilni proces. Tri najbolj razširjene plačilne sisteme, ki uporabljajo tehnologijo NFC bomo predstavili v nadaljevanju diplomskega dela.

Še pred tem bomo predstavili in raziskali zmožnosti tehnologije NFC. V drugi polovici bomo skozi načrtovanje in samo implementacijo demonstrirali uporabo tehnologije NFC v mobilni aplikaciji na primeru turističnega vodiča. Predstavljena aplikacija bo razvita za operacijski sistem Android, zaradi njegove trenutne razširjenosti. V drugem četrtletju leta 2013 je bilo kar 79 odstotkov mobilnih naprav s pred nameščenim operacijskim sistemom

Android [1]. Kot kažejo trendi, se bo tržni delež teh naprav v prihodnosti še povečal, kar predstavlja odlično priložnost za razvoj in trženje tovrstnih aplikacij.

# Poglavje 2

## Near field communication

Near field communication (NFC) je brezstičen standard, ki deluje na kratkih razdaljah s pomočjo radijskih signalov. Zaradi svojih lastnosti postaja vse bolj razširjen predvsem v mobilnih sistemih. Prednost v primerjavi z ostalimi brezžičnimi standardi je, poleg nizke porabe električne energije, predvsem kratek vzpostavitevni čas povezave. Trenutno najbolj aktualno področje uporabe je uporaba v mobilnih aplikacijah, predvsem mobilno plačevanje.

### 2.1 Razvoj

Na nastanek standarda NFC je vplivala predvsem potreba po varnosti. Tako se je standard razvil iz že dolgo obstoječega standarda RFID. Za hiter napredek razvoja pa skrbi predvsem organizacija NFC forum.

#### 2.1.1 RFID

Standard NFC je nekakšna izpeljanka že dolgo obstoječega standarda RFID. Slednjega so prvič uporabljali že med drugo svetovno vojno pri radarskih sistemih za razlikovanje med sovražnikovimi in zavezniškimi letali. Radar je oddajal radijski signal, na katerega je radijski sprejemnik na letalu ustrezno odgovoril. Tehnologija je bila izumljena že leta 1948, vendar se je za komercialne namene začela uporabljati šele v osemdesetih letih devetnajstega

stoletja.

V današnjem času se uporablja predvsem pri sledenju oziroma označevanju izdelkov v trgovinah, skladiščih, itd. Tako uporabo omogoča predvsem nizka cena elektronskih vezij, kateri sestavljajo sprejemnike in oddajnike RFID. V zadnjem času cena teh vezij hitro dosega magično mejo 5 dolarskih centov [2]. Le-ta omogoča trgovcem, da označujejo tudi izdelke nižjih vrednosti (npr. pijače). Sama tehnologija pa trgovcem močno zmanjša stroške skladiščenja in prevoza, saj imajo vseskozi na voljo aktualne podatke o izdelkih. Močno se zmanjšajo tudi stroški in čas zaračunavanja izdelkov. Vse kar mora kupec storiti je to, da zapelje voziček poln izdelkov označenih z značko RFID skozi oddajnik na blagajni. Blagajnik ima tako praktično v trenutku vse potrebne podatke za izdajo računa. Eni izmed prvih, ki so zaupali tej tehnologiji so bili pri ameriški trgovski verigi Walmart skupaj z njenimi dobavitelji.

Uporaba RFID je torej precej podobna črtni kodi, le da izdelke namesto s tiskano kodo - UPC označujemo z elektronsko - EPC. S tem pridobimo predvsem na avtomatizaciji procesa prodaje, saj za prebiranje črtnih kod običajno potrebujemo prodajalce, ki le-to približajo čitalcu. Velika pohitritev je tudi število prebranih zapisov, saj se pri RFID to meri v več 100 značk hkrati, medtem ko v tem času lahko preberemo le eno črtno kodo. Poleg tega nam EPC omogoča naknadno spreminjanje podatkov. Doseg RFID tehnologije je običajno nekaj metrov, ob pravih pogojih pa standard omogoča tudi do 200 metrov dosega. Za namene uporabe na kratke razdalje in povečanja varnosti se je iz tehnologije RFID razvila tehnologija NFC [2, 3].

### **2.1.2 NFC forum**

NFC forum je neprofitno združenje podjetij, ki je bil ustanovljen leta 2004 z namenom pohitritve razvoja in vpeljave tehnologije NFC. V okviru foruma se razvijajo specifikacije, ki zagotavljajo interoperabilnost med napravami in storitvami. Njihova naloga je tudi spoznavanje širše javnosti s tehnologijo. Trenutno šteje več kot 170 članov, med njimi tudi Google Inc., Broadcom Corporation, QUALCOMM Inc., Intel Corporation, Samsung, Sony, Visa,

MasterCard, NXP Semiconductors in drugi.

Do danes je na tem področju izdalo že 16 specifikacij [4].

## 2.2 Delovanje

NFC deluje podobno kot kreditne kartice, s pomočjo magnetne indukcije med dvema napravama. Naprava, ki želi vzpostaviti komunikacijo oddaja električen tok na radijsko-frekvenčnem spektru 13.56MHz. Pri tem porabi manj kot 15mA električne energije, zaradi česar je standard energijsko zelo varčen. Posledica tega je magnetno polje med napravama, kjer druga naprava odgovori z električnimi impulzi in tako prenese podatke v nasprotno smer. Naprave NFC ločimo v dve skupini:

- Pasivne so tiste, ki niso napajane z električno energijo in tako tudi ne morejo same vzpostaviti magnetnega polja. Za komuniciranje potrebujejo aktivno napravo, da ustvari magnetno polje iz katerega se pasivne naprave napajajo. Tipičen predstavnik je značka NFC (ang. NFC tag).
- Aktivne naprave so zmožne ustvariti magnetno polje, torej lahko berejo in pišejo podatke. Med dvema aktivnima napravama lahko poteka dvosmerna komunikacija, kjer naprava, ki prejema podatke deluje v pasivnem načinu. Torej, vedno oddaja le ena aktivna naprava. Najbolj razširjen predstavnik je mobilni telefon.

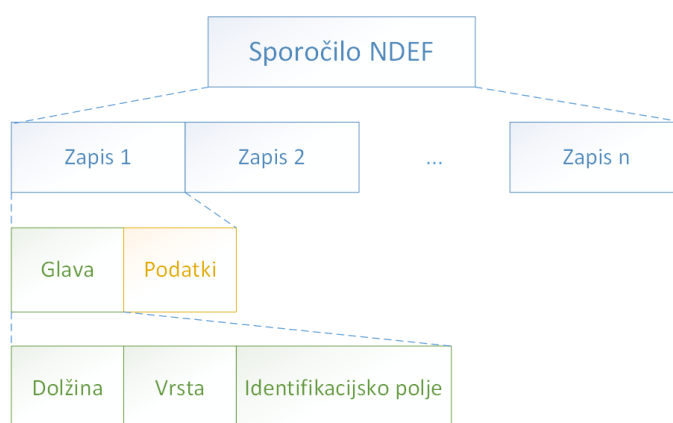
### 2.2.1 NFC značka

Delovanje značk NFC je zelo podobno znački RFID, le da so podatki na značkah NFC zapisani v ustreznem formatu, ki je berljiv vsem aktivnim napravam NFC. Značke delimo v 4 različne razrede, glede na velikost podatkovnega prostora in hitrosti prenosa podatkov, našteje v tabeli 2.1. Priznani so s strani združenja NFC Forum in posledično združljive z vsemi NFC Forum priznanimi napravami.

Razred značke NFC	Podatkovni prostor	Hitrost prenosa podatkov
Tip 1	96B - 2KB	106 kbit/sec
Tip 2	96B - 2KB	106 kbit/sec
Tip 3	2KB	212 kbit/sec
Tip 4	32 KB	106 - 424 kbit/sec

Tabela 2.1: Štirje tipi značk NFC [5]. Prva dva lahko ponovno zapišemo in tudi zaščitimo pred nadaljnjimi pisanji, med tem ko zadnjih dveh ne moremo zapisovati. Podatki so običajno na značke shranjeni že med proizvodnjo ali preko posebnih zapisovalcev.

Sporočila na značkah so lahko zapisani v različnih formatih. Eden izmed najbolj uporabljenih je format NDEF, ki ga je uvedlo združenje NFC forum. Sporočila NDEF (ang. NDEF message) so sestavljena iz enega ali več zapisov NDEF (ang. NDEF record). Koliko teh zapisov vsebuje samo sporočilo je odvisno od tipa značke in aplikacije, ki sporočilo uporablja. Vsak zapis vsebuje glavo in podatke. Pri čemer glava vsebuje za aplikacijo potrebne informacije o podatkih: dolžina podatkov v bajtih, vrsta podatkov in identifikacijsko polje (neobvezno). Sestava sporočila je prikazana na sliki 2.1.

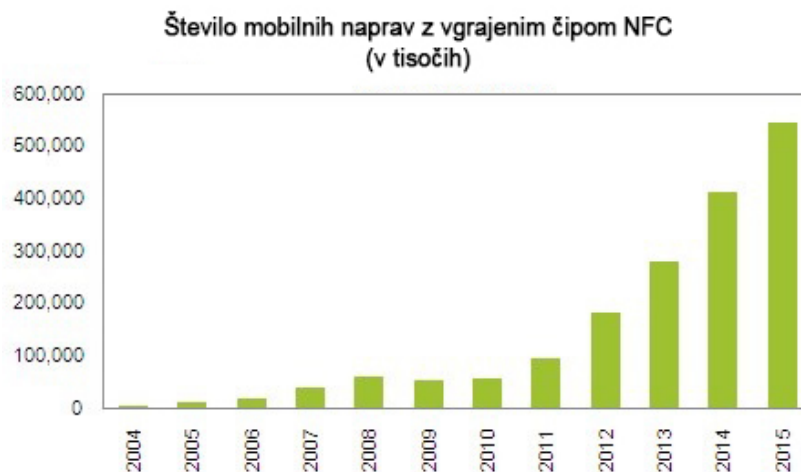


Slika 2.1: Sestava zapisa na znački NFC po standardu NDEF.

Podatki so lahko različnih vrst, običajno je to naslov URI ali eden izmed tipov MIME (text, slika, zvok, itd.). Večina mobilnih operacijskih sistemov ima razvite APIje, ki znajo prebrati sporočila NDEF različnih vrst.

### 2.2.2 Mobilna naprava kot aktivna NFC naprava

Vedno več operacijskih sistemov podpira standard NFC. Poleg Androida so to še Blackberry OS, Windows, Symbian, Bada in drugi. Zato je pomembno da so podatki zapisani po nekem standardu, kot je že omenjeni NDEF, da lahko iste značke prebirajo različne mobilne naprave. Po raziskavah različnih podjetij bo število naprav z vgrajenim čipom NFC strmo naraščalo. Po napovedih tržno-raziskovalnega podjetja IHS iSuppli (slika 2.2) naj bi bilo leta 2015 dobrih 500 milijonov mobilnih naprav z vgrajenim čipom NFC. Ne gre zanemariti tudi, da se Apple do današnjega dne v svoje naprave še ni odločil vgraditi čipa NFC.



Slika 2.2: Naraščanje mobilnih naprav z vgrajenim čipom NFC. Raziskava je bila narejena maja 2011 s strani tržno-raziskovalnega podjetja IHS iSuppli [6].

Aktivna naprava lahko spremeni stanje v pasivno, kar omogoča dvo-

smerno izmenjevanje podatkov med dvema napravama. To odpira številne možnosti za implementacijo novih aplikacij za povezljivost in izmenjevanje podatkov med napravami, kar trenutno poteka preko za to razvitega standarda Bluetooth, ali kar preko energijsko potratnega Wi-Fi. Prednost omenjenih standardov v primerjavi z NFCjem je veliko višja hitrost prenosa podatkov.

Ker ima NFC veliko hitrejši vzpostavitevni čas od preostalih in počasnejšo hitrost prenosa podatkov, se razvijajo nove tehnologije, ki združujejo prednosti omenjenih standardov. Ena izmed teh je Android Beam, ki je bil predstavljen v različici operacijskega sistema Android 4.0. V prvotni implementaciji so se tudi podatki prenašali preko NFC, v različici Androida 4.1 pa so predstavili novo implementacijo, kjer se NFC uporabi za vzpostavitev in aktivacijo protokola Bluetooth na obeh napravah za prenos podatkov.

Še korak dlje so šli pri podjetju Samsung, ki so predstavili S Beam, nadgradnjo tehnologije Android Beam, kjer se za prenos podatkov uporablja kar protokol Wi-Fi, ki omogoča še višjo hitrost prenosa od protokola Bluetooth.

## 2.3 Varnost in vrste možnih napadov

Kot pri vseh brezžičnih tehnologijah se tudi pri tehnologiji NFC poraja vprašanje varnosti. Pri slednji še bolj, predvsem zaradi uporabe pri mobilnem plačevanju. Poznamo najmanj štiri možne napade na tehnologijo NFC [7, 2]: prisluškovanje, okvara in manipulacija podatkov, prestrezanje povezave in kraja naprave. V naslednjem delu je predstavljena vsaka kategorija in možne rešitve, kako se takim nevarnostim izogniti ali vsaj zmanjšati možnost pojava.

### 2.3.1 Prisluškovanje

Pri tem napadu napadalec prisluškuje dani transakciji NFC in pridobi potrebne informacije za izvedbo nadaljnjega napada, recimo podatke o kreditni kartici pri plačevanju z mobilno napravo.

Glavna težava za napadalca pri tem je že sama fizična omejitev standarda NFC, ki omogoča povezave na razdalji do 20 centimetrov. Običajno je ta razdalja še precej manjša, kar pomeni da ima napadalec zelo malo fizičnega prostora za izvedbo napada.

Poleg tega standard NFC definira tudi vzpostavitev varnostnih kanalov za prenos podatkov. Podatki se tako prenašajo šifrirano in samo pooblašcene naprave lahko podatke pretvorijo v čistopis. Uporabnik naj bo pri plačevanju s tehnologijo NFC pozoren na to, da podjetja, ki ponujajo brezstično plačevanje uporabljajo varnostne kanale.

### **2.3.2 Okvara in manipulacija podatkov**

Okvara in manipulacija se zgodi, ko napadalec spremeni podatke, ki so bili poslani proti bralniku NFC ali moti povezavo in s tem okvari podatke, ki tako postanejo neuporabni. Poleg naprav, ki zaznavajo okvare in spremembe podatkov so rešitev proti takim napadom že omenjeni varnostni kanali.

### **2.3.3 Prestrežanje povezave**

Napad je znan tudi pod angleškim izrazom man-in-the-middle, kjer se napadalec vrine med izvor in ponor povezave ter tako nanjo neposredno vpliva. Napadalec prejme vse informacije, jih lahko spremeni in posreduje naprej. Sama izvedba je zaradi delovanja standarda NFC na kratkih razdaljah fizično zelo otežena in zato tudi redka.

### **2.3.4 Kraja naprave**

Ta napad je verjetno za uporabnika največja nočna mora. Napadalec teoretično lahko z ukradenim telefonom plačuje račune v vašem imenu. Takim neprijetnim dogodkom se je najlažje izogniti s tem, da zaščitimo telefon z geslom ali katero koli drugo metodo zaklepa telefona, ki napadalcu ne omogoča neposredne uporabe vaših podatkov.

Če povzamemo varnostna vprašanja in dileme o standardu NFC, lahko vidimo, da je dobro zasnovan in napadalcu zelo otežuje pridobivanje in izkoriščanje uporabnikovih podatkov. Obstajajo tudi druge nevarnosti, ki pa se ne tičejo toliko samega standarda. Tukaj bi omenil predvsem izmenjevanje zlonamerne kode, kar je pa bolj vprašanje aplikacijskega nivoja, kaj narediti s pridobljenimi podatki, kot pa samega standarda NFC.

V primerjavi s kreditno kartico, katero naj bi standard v bodoče nadomestil, lahko zaključimo, da je NFC dovolj varen, če ne celo še bolj varen od kreditnih kartic. Če primerjamo največje tveganje, kraja NFC naprave ali kreditne kartice, imamo pri prvi več možnosti da se zaščitimo proti nadaljnji uporabi, med drugim lahko napadalca celo izsledimo. Poleg varnosti pa ne smemo pozabiti še drugih prednosti tehnologije NFC proti kreditni kartici. Ni nam potrebno nositi več ločenih kartic, ampak imamo vse shranjene na eni napravi, postopek je hitrejši in tako dalje.

## **2.4 Razširjenost danes in možnosti za prihodnost**

Predstavljeno je že, da je največji potencial mobilno brezstično poslovanje, sploh iz poslovnega vidika. Glavni razlog za trenutno nerazširjenost takšnega poslovanja je pomanjkanje infrastrukture. Tu gre predvsem za pomanjkanje strojne opreme na prodajnih mestih in pa preverjen, učinkovit ter varen informacijski sistem. Ni pa mobilno brezstično poslovanje edini način uporabe tehnologije NFC, zato si bomo v nadaljevanju pogledali tudi ostale možne uporabe.

### **2.4.1 Mobilno kontaktno poslovanje**

Kot smo videli število mobilnih naprav strmo narašča (slika 2.2), vendar je delež naprav z vgrajenim NFC še vedno zelo nizek. Tako tudi ni mogoče z ra-

znimi storitvami NFC zadovoljiti dovolj ljudi, kar pomeni implementacijo več različnih načinov plačila na prodajnih mestih in posledično dodaten strošek. Vendarle so vodilna podjetja zaznala, da je delež takih naprav že dovolj visok za prve implementacije informacijskih sistemov na tem področju. S tem naj bi predvsem poenostavili in poenotili plačevanje preko spleta ter običajne načine plačevanja. V naslednjih letih lahko pričakujemo strm porast števila brezstičnih plačil [8].

### Visa payWave

Implementacija Visinega brezstičnega plačevanja, ki so ga predstavili že leta 2007. Namenjen je uporabi prilagojenih plačilnih kartic z zamahom blizu terminala, brez da jih potegnemo skozi POS terminal. Kartice so označene s posebnim simbolom prikazanem na sliki 2.3. Kartice imajo vgrajeno anteno za delovanje po standardu RFID in so označene s posebnim simbolom. Delovanje kartic je omogočeno na razdalji desetih centimetrov. Za plačila manjših vsot (npr. do 20€) se transakcija izvede takoj, ob višjih vsotah se uporabi standardna metoda overjanja z vnosom identifikacijske številke PIN.



Slika 2.3: Znak, ki označuje plačilne kartice s sposobnostjo brezstičnega plačevanja Visa payWave [9].

Tu pa nastopijo tudi mobilne naprave z vgrajenim čipom NFC. Ker je standard NFC izpeljan iz standarda RFID (razdelek 2.1.1) se pri plačevanju lahko uporabljajo tudi take mobilne naprave. Konec februarja 2013 sta Visa in Samsung, kot vodilni proizvajalec mobilnih naprav, podpisala pogodbo [10], po kateri bodo naprave Samsung vsebovale ustrezno programsko opremo

za uporabo storitve Visa payWave. S tem naj bi pospešila uporabo mobilnega plačevanja.

### **MasterPass**

MasterPass je še ena plačilna storitev z velikimi načrti v prihodnosti na področju brezstičnega mobilnega plačevanja. Razvija jo ameriško podjetje MasterCard in je evolucija že obstoječe storitve PayPass Wallet. Uporabnik si v varnem oblaku lahko shrani osebne in bančne podatke. S sistemom so povezane digitalne denarnice, ki jih lahko svojim strankam izdajajo banke in trgovine. Trenutno že precej bank podpira uporabo storitve, kot recimo Banco Santander, BBVA Bank, BNP Paribas Fortis Belgium, UniCredit in druge. In seveda tudi večji trgovci: Harvey Norman, American Airlines, Vodafone Italy in drugi. Podpira vse večje bančne kartice: MasterCard, AmericanExpress, Visa, DinersClub International in Discover Network [11]. Vsa ta partnerstva zaenkrat dajejo prednost storitvi MasterPass v primerjavi s storitvijo Visa payWave. Storitev prepoznamo po znaku 2.4.



Slika 2.4: Znak po katerem bomo prepoznali storitev MasterPass na plačilnem mestu [11].

Sam način plačevanja v trgovini je precej podoben postopku opisanemu v Visini storitvi (razdelek 2.4.1).

### **Google Wallet**

Virtualna Googleova denarnica je brezplačna storitev za plačevanje tako preko spleta, kot tudi v trgovinah. Prav tako kot že predstavljena predstavnika je tudi Google Wallet prepoznan po svojem znaku, na sliki 2.5. V nadaljevanju se bomo osredotočili na mobilno aplikacijo, ki je del te storitve

in nam omogoča prenos fizičnih plačilnih kartic v mobilno napravo. Google Wallet nam omogoča, da na mobilno napravo in Googleov oblak varno shranimo naše plačilne kartice in tudi razne promocijske kode, katere kasneje uporabimo pri plačilu. Od sredine maja 2013 nam storitev omogoča celo plačevanje preko Googleove spletne pošte Gmail. Aplikacija je trenutno na voljo le v Združenih državah Amerike (oziroma napravah kupljenih v tej državi), kjer je na voljo že od konca leta 2011, vendar samo za osebe starejše od 18 let. V nekaterih ostalih državah storitev lahko uporabljamo za plačevanje preko spleta.



Slika 2.5: Zaščitni znak storitve Google Wallet [12].

Plačevanje preko NFC je omogočeno na vseh terminalih PayPass (predhodnik storitve MasterPass), kjer samo približamo telefon terminalu NFC in izberemo vir plačila. Poleg varnosti NFC (razdelek 2.3) so pri Googleu poskrbeli tudi za varnost na aplikacijskem nivoju in tako sama aplikacija ob uporabi zahteva PIN. Poleg tega se uporabnikovi podatki pred shranjevanjem šifrirajo. V primeru izgubljene naprave lahko aplikacijo oddaljeno izklopimo preko Googleovega računa.

### 2.4.2 Ostala področja uporabe

Poleg brezstičnega poslovanja in prenosa podatkov med mobilnimi napravami se tehnologija NFC uporablja, predvsem z značkami NFC, za različne namene:

- Vklon funkcij med vožnjo: ko se usedemo v avto napravo položimo na značko, ki nam izklopi WiFi in vklopi Bluetooth za prostoročno telefoniranje ali za predvajanje glasbe.
- Vklon funkcij na delovnem mestu: na delovnem mestu vklopimo WiFi, vklopimo tihi način...
- Avtomatiziran dostop do zaščenega WiFi omrežja: napravo približamo znački NFC, ki sama sproži overjanje z usmerjevalnikom omrežja, brez da bi vnašali gesla in iskali omrežje.
- Vklon računalnika ob dotiku: Ob dotiku značke NFC, naprava sproži samodejen vklop osebnega računalnika.
- Avtomatizacija domačih opravil: Samodejen vklop in izklop luči v domu, odpiranje vrat in tako dalje.
- Zagon aplikacij: ob dotiku značke NFC se na operacijskem sistemu zažene aplikacija.
- Varčevanje z električno energijo: na mestih s slabim mobilnim signalom izklopimo mobilno omrežje, WiFi omrežje omogočimo le na za to namenjenih mestih...
- Nadomeščanje kode QR: na promocijskih plakatih nadomestimo kode QR s poceni značkami NFC. Nanje lahko shranimo več podatkov, predvsem pa pridobimo na hitrejšem odzivu.

Značke NFC se torej uporabljajo predvsem za označevanje lokacij. V nadaljevanju je predstavljena aplikacija, ki na preprostem primeru turističnega vodiča demonstrira uporabo tehnologije NFC. Značke NFC so uporabljene za označevanje posameznih znamenitosti.

## Poglavje 3

# Tehnologije in oprema uporabljena pri razvoju aplikacije

V tem poglavju so predstavljene tehnologije in opreme, ki so potrebne za razvoj mobilne aplikacije in tudi strežniškega dela sistema. Zato so v prvem delu predstavljene tehnologije povezane z mobilno aplikacije, od vključno storitev REST naprej pa še tehnologije na strani strežnika.

### 3.1 Android

Android je odprtokodni mobilni operacijski sistem, ki je osnovan okoli Linu-xovega jedra. Prvotno so ga razvijali pri podjetju Android, Inc., katerega je v tem času finančno podpiral spletni gigant Google. Slednji je prvotno podjetje kupil leta 2005 in ga trenutno tudi aktivno razvija.

#### 3.1.1 Označevanje različic

Zadnja izdana različica je različica 4.3, razvita pod imenom Jelly Bean (enako ime se uporablja tudi za verziji 4.1 in 4.2). Vsaka večja izdaja ima svoje ime, npr. Jelly Bean, Ice Cream Sandwich, Honeycomb,...

Z vsako Android različico Google izda tudi nov programski vmesnik (API), ki so oštevilčeni s številkami od 1 naprej. API prve generacije je pripadal Androidu 1.0, zadnji API, 18. generacije, pripada verziji 4.3. V tabeli 3.1 so našteje najpogosteje uporabljene različice operacijskega sistema Android. Vse različice, ki imajo razširjenost manjšo od 0.1% so izvzete, predvsem starejše od različice 2.2.

Različica	Ime	API	Razširjenost
2.2	Froyo	8	2.4%
2.3.3 - 2.3.7	Gingerbread	10	30.7%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	21.7%
4.1.x	Jelly Bean	16	36.6%
4.2.x		17	8.5%

Tabela 3.1: Podatki o razširjenosti različic Android zbranih v časovnem obdobju 14. dni, končanih 4. septembra, 2013 [13].

Prvi API, ki prinaša podporo za tehnologijo NFC je 10. generacija oziroma Android 2.3.3. Iz istega razloga, je aplikacija, predstavljena v nadaljevanju, razvita le za telefone, kateri uporabljajo Android 2.3.3 ali novejši. Vse podobne nastavitve se pri Androidu nastavljajo v glavni datoteki XML, `AndroidManifest.xml`, prikazano na primeru 3.1.

Listing 3.1: Defniramo najnižjo in optimalno podprto verzijo Androida

```

1 <uses-sdk
2     android:minSdkVersion="10"
3     android:targetSdkVersion="18" />

```

### 3.1.2 Zgradba aplikacij

Vsaka Android aplikacija vsebuje osrednjo datoteko `AndroidManifest.xml` v kateri so definirane nastavitve aplikacije. Poleg verzij Androida se lahko določijo tudi potrebne sistemske pravice, vezane programske knjižnice, in najbolj pomembno, definicije gradnikov in odvisnosti med njimi. Vsaka aplikacija v sistemu Android je sestavljena iz ene ali več od štirih možnih komponent:

- Aktivnosti (ang. Activity): Edina možnost, da aplikacijo poženemo z uporabniškega vmesnika. Sistem omogoča preklapljanje med različnimi aktivnostmi. Običajno imajo kratke življenske cikle, kar je podrobneje opisano kasneje.
- Storitve (ang. Service): Namenjene so dolgemu življenskemu ciklu in so (lahko) neodvisne od aktivnosti. Nenehno izvajajo proces v ozadju.
- Upravitelj vsebine (ang. Content provider): Komponenta s katero se preko vmesnika definira dostop ostalih aplikacij (lahko tudi svoji) do podatkov naše aplikacije.
- Prejemnik sporočil (ang. Broadcast receiver): Prejema sporočila, ki so poslana iz drugih aplikacij ali samega sistema. Sporočila se običajno označujejo kot dogodki (ang. events) ali namere (ang. intents).

### 3.1.3 Aktivnost in njen življenski cikel

V aplikaciji, predstavljeni v nadaljevanju, bo kot edina od naštetih gradnikov uporabljena aktivnost. Slednja je glavna komponenta vseh aplikacij operacijskega sistema Android. Vsaka aktivnost je lahko v enem od naslednjih stanj [14]:

- aktivna (ang. active): Postavljena v ospredje, odziva se na uporabniške interakcije.

- prekinjena (ang. *paused*): Ni postavljena v ospredje, je pa še vedno delno vidna in se ne odziva na uporabnikove interakcije (npr. pred njo je na zaslonu nek dialog).
- ustavljena (ang. *stopped*): Aktivnost na zaslonu ni več vidna, vendar je še vedno shranjena v delovnem spominu sistema.
- uničena (ang. *destroyed*): Sistem je aplikacijo izbrisal iz delovnega spomina, za ponovni zagon je potreben klic metode `onCreate()`.

Aktivnost lahko prehaja med stanji, katere se lahko preko določenih metod zazna programsko. Kot edina obvezna metoda, katero moramo implementirati je metoda `onCreate`. Metoda se izvede vedno, ko operacijski sistem znova zgradi uporabniški vmesnik - prvi zagon aplikacije, obrnjen položaj zaslona... Prehodi med stanji so predstavljeni na diagramu 3.1.

## 3.2 Java

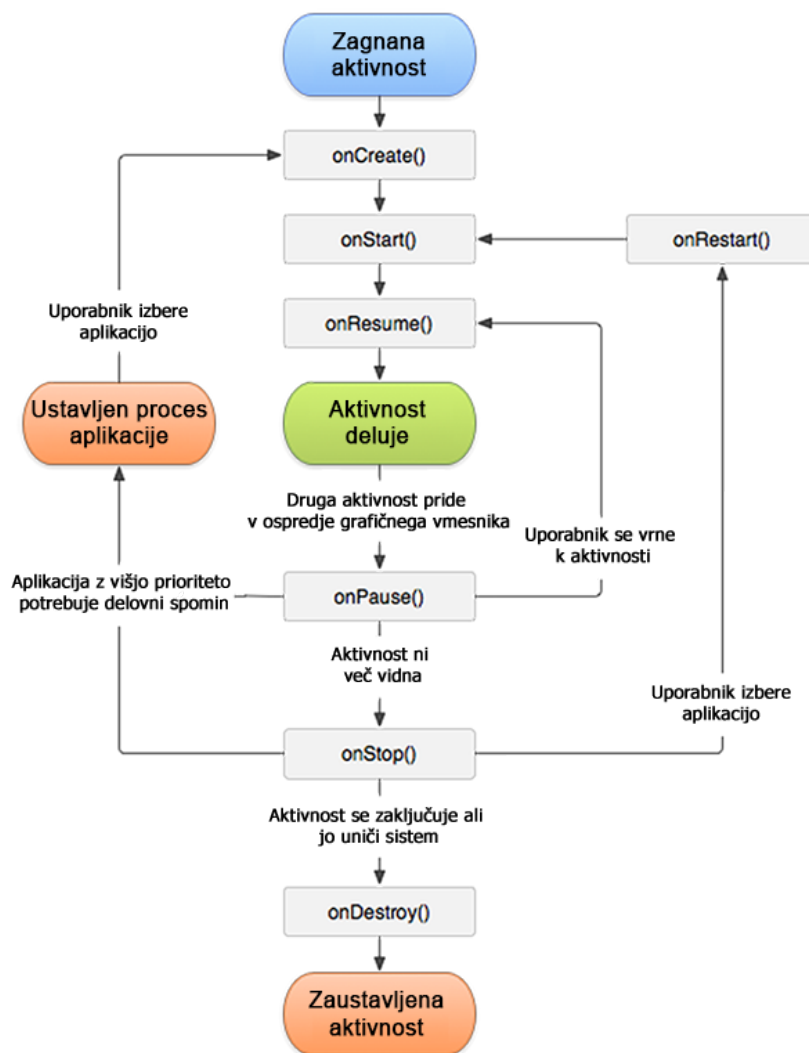
Aplikacija je implementirana s pomočjo programskega jezika Java. Slednji je trenutno najbolj razširjen programski jezik izmed vseh [16].

Na uradni strani razvijalcev Android je prosto dostopen paket SDK, ki vsebuje vse potrebne APIje in druge knjižnice za razvoj mobilnih aplikacij Android v jeziku Java. V primeru 3.2 je prikazana preprosta uporaba programskega jezika Java za izpis besedila na standardni izhod.

Listing 3.2: Demonstracija uporabe programskega jezika Java s preprostim izpisom 'Pozdravljen svet!' na standardni izhod.

```
1 public class MojTestniRazred{
2     public static void main(String[] args) {
3         System.out.println("Pozdravljen_svet!");    //Komentar: izpis
3             na standardni izhod
4     }
5 }
```

---



Slika 3.1: Življenski cikel aktivnosti v Androidu, s potrebnimi metodami za prehajanje med stanji [15].

### 3.3 Android Studio

Razvojno okolje za razvijanje Android aplikacij. Prva različica je bila predstavljena na konferenci Google I/O 2013. Razvojno okolje temelji na že uveljavljenem razvojnem okolju IntelliJ IDEA. Prinaša sveže pristope za razvijanje uporabniškega vmesnika aplikacij na več različnih zaslonih in optimizirane algoritme za organiziranje aplikacijske izvorne kode. Prav tako vsebuje vgrajen paket ADT za testiranje aplikacije v posnemovalniku (ang. emulator).

### 3.4 Testno okolje

Poleg omenjenega posnemovalnika je bila aplikacija testirana tudi s pomočjo dodatne programske opreme GenyMotion, ki namesto posnemovalnika Android naprave (ang. emulator) uporablja virtualizacijo s pomočjo programske opreme VirtualBox in s tem dosega občutno hitrejšo hitrosti zagona aplikacije. Za končno testiranje in testiranje delovanja interakcije aplikacije z značkami NFC je bila uporabljena mobilna naprava z nameščenim operacijskim sistemom Android 4.3, Samsung Galaxy Nexus i9250.

### 3.5 SQLite

Za shranjevanje podatkov lokalno, na mobilni napravi, je uporabljena v Android vgrajena podatkovna baza SQLite. Slednja je prostorsko in procesorsko učinkovita, za poizvedovanje pa se uporablja poenostavljen poizvedovalni jezik SQL. Široko se uporablja v vgrajenih sistemih (ang. embedded systems) in brskalnikih. Dostop do baze iz aktivnosti je izvršen preko razvitega programskega vmesnika `MySQLiteHelper.java`, kjer sem definiriral vse potrebne metode za upravljanje s podatki v podatkovni bazi.

## 3.6 Storitve REST

Ena izmed možnosti arhitekture šibko sklopljenih sistemov so storitve REST. Svojo razširjenost so dosegle z razvojem svetovnega spleta. Arhitektura je bila razvita vzporedno s standardom HTTP 1.1 v sklopu mednarodne organizacije W3C.

Kot osnova arhitekture je standard HTTP, nadalje pa lahko arhitekturo ločimo na dva pomembna člena: odjemalca in strežnika. Odjemalec preko spletnega naslova odda zahtevo, katero strežnik obravnava in pošlje nazaj odgovor odjemalcu. Odgovor je lahko v več različnih oblikah: JSON, XML, slika, video,... Vsebina samih odgovorov so podatki iz podatkovnih baz strežnika zapisani v naštetih oblikah.

Taka arhitektura povečuje varnost sistemov, saj s tem uporabnikom storitve omejimo vidnost strežniških podatkov. Pridobimo na neodvisnosti različnih sistemov in skalabilnosti sistemov.

Druga podobno razširjena arhitektura je SOAP. Slednja popolnoma temelji na razširjenem standardu XML, kateri definira same storitve preko WSDL vmesnika, sestavlja odgovore,...

## 3.7 JSON

Je preprost format za zapis in izmenjavo človeško in strojno berljivih podatkov. Temelji na podmnožici programskega jezika JavaScript, Standard ECMA-262, tretja izdaja - december 1999 [17]. Kljub temu pa je splošno podprt tudi v ostalih programskih jezikih. Njegova največja prednost je prostorska učinkovitost in kompaktnost zapisa podatkov v primerjavi s sorodnim standardom XML. Predstavlja preprost zapis programskih objektov v berljivo strukturo, kot je prikazano na primeru 3.3.

Listing 3.3: Primer JSON objekta, kjer imamo v polju 'filmi' shranjene podatke o posameznih filmih.

```
2  "filmi": [  
3    { "naslov":"Umri_pokoncno" , "zvrst":"akcija" },  
4    { "naslov":"Titanik" , "zvrst":"drama" },  
5    { "naslov":"Zaga_5" , "zvrst":"grozljivka" }  
6  ]  
7 }
```

---

JSON, uporabljen pri REST storitvah je sintaktično preverjen s pomočjo spletnega preverjevalnika JSONLint.

## 3.8 PHP

PHP je preprost skriptni programski jezik, ki se običajno izvaja na spletnem strežniku. Ustvarjen je bil z namenom izdelave domačih, dinamičnih spletnih strani in se razširil v enega izmed najbolj popularnih programskih jezikov. Programska koda se interpretira s pomočjo procesorja PHP in običajno vrača rezultate v zapisu HTML, ko gre za spletne strani, oziroma JSON ali XML, ko gre za spletne storitve. Med drugim se s programskim jezikom PHP vrši dostop do podatkovne baze MySQL. Primer skriptnega jezika PHP najdemo v preprostem primeru 3.4, gre za preprost izpis v telo dokumenta HTML.

Listing 3.4: Primer vključevanja PHP kode v kodo HTML.

```
1 <html>  
2 <head />  
3 <body>  
4 <?php  
5   //To je komentar v odseku kode PHP  
6   echo "<p>Pozdravljen_svet!</p>  
7 ?>  
8 </body>
```

---

## 3.9 MySQL

MySQL je odprtokoden sistem za upravljanje z relacijskimi podatkovnimi bazami. Za razvoj in vzdrževanje skrbi podjetje Oracle. Sistem podpira vse večje operacijske sisteme in se zlahka namesti preko uradne spletne strani. Za poizvedovanje po podatkih v podatkovni bazi se uporablja poizvedovalni jezik SQL, katerega uporaba je predstavljena na primeru 3.5.

Listing 3.5: Preprost primer poizvedovalnega jezika SQL ki nam vrne vse podatke o slovenskih dvatisočakah

```
1 SELECT * FROM gore WHERE nadmorska_visina > 2000 AND drzava = '
   Slovenija'
```

Med drugim podpira tudi različne pogone za shranjevanje podatkov (ang. storage engine), kot sta recimo InnoDB, ki je sistemsko privzet od različice MySQL 5.5 in MyISAM. Prvi je namenjen za učinkovite transakcije, drugi optimizira hitrost branja podatkov. Za dosego višje učinkovitosti aplikacije ima lahko vsaka tabela različen pogon za shranjevanje.

Med razvojem aplikacije sem si pomagal s spletnim vmesnikom phpMyAdmin za dostop do podatkovne baze. Slednji temelji na že omenjenem jeziku PHP. Omogoča pregled podatkov in upravljanje same podatkovne baze preko spletnega grafičnega vmesnika.



## Poglavje 4

# Načrtovanje in implementacija mobilne aplikacije

V nadaljevanju bosta predstavljena načrtovanje in implementacija aplikacije, ki demonstrira uporabo tehnologije NFC. Za večjo prepoznavnost je poimenovana NFC Traveler in bo lahko prepoznana po logotipu na sliki 4.1. Aplikacija bo turističen vodič, katerega znamenitosti bodo enolično označene s posamezno značko NFC, ki nam bo služila za označevanje lokacij. Uporabnik aplikacije bo lahko potrdil obisk znamenitosti in si hkrati ogledal podatke o obiskani znamenitosti. Lahko pa bo samo prebiral seznam vseh znamenitosti in si ogledal njihove podatke. Podatki vsebujejo podrobnosti, kot je opis in število vseh obiskov znamenitosti ter samo sliko znamenitosti. Podprte bodo glavne znamenitosti glavnega mesta Slovenije, Ljubljane, vendar nam aplikacija v prihodnje omogoča preprosto širitev.

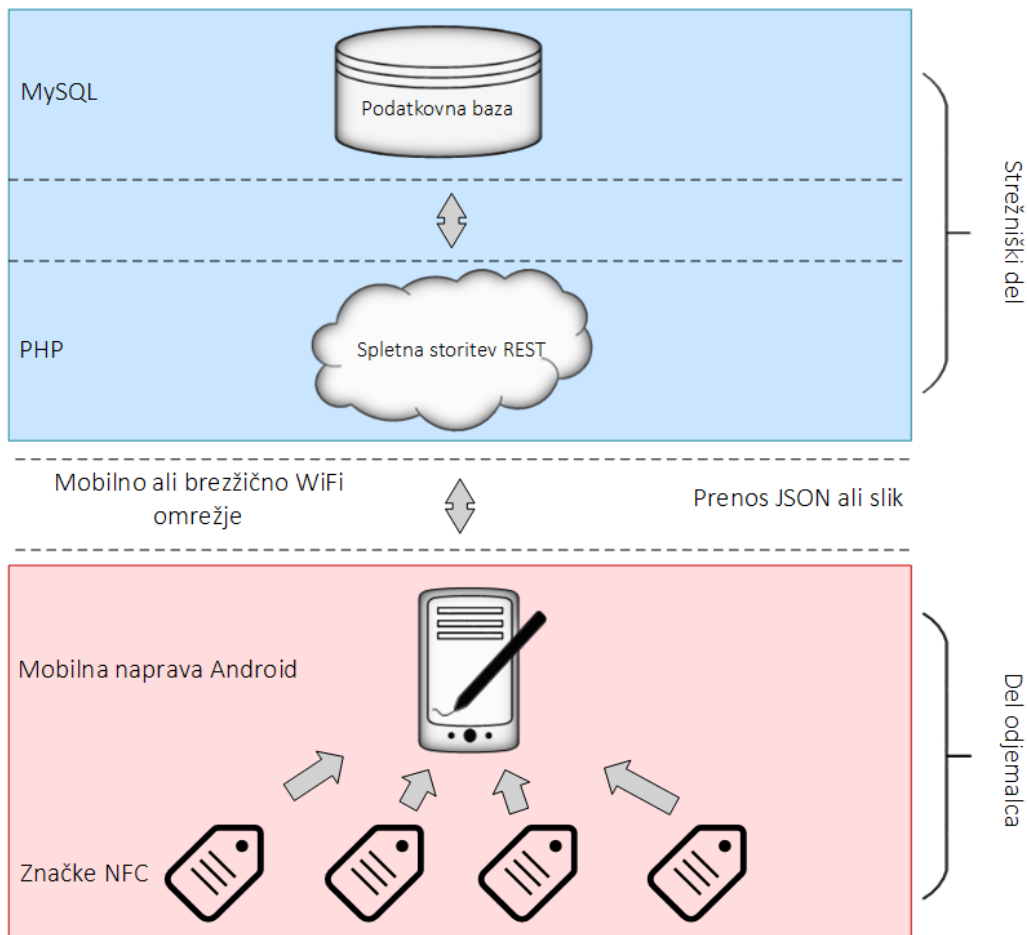
Aplikacija podpira dva jezika - slovenskega in angleškega. Zato je razvita v skladu s standardi operacijskega sistema Android, tako da so vsa besedila mobilne aplikacije shranjena v datoteki `strings.xml`, kar nam omogoča hiter prevod. V ta namen je prilagojen tudi strežniški del.



Slika 4.1: Logotip aplikacije NFC Traveler. Rjav pravokotnik simbolizira potovalni kovček, simbol znotraj njega pa tehnologijo NFC.

## 4.1 Načrtovanje celotnega sistema

Na diagramu 4.2 je predstavljen načrt celotnega sistema. Sistem je ločen na strežniški del, katerega osrednji del so spletne storitve REST in del odjemalca, katerega osrednji del je mobilna aplikacija. Povezava med deloma pa je internetna povezava preko katere se prenašajo slike znamenitosti ali podatki v obliki zapisa JSON. Na diagramu so razvidne tudi uporabljene tehnologije pri posameznem delu sistema, ki so podrobneje opisane v poglavju 3.



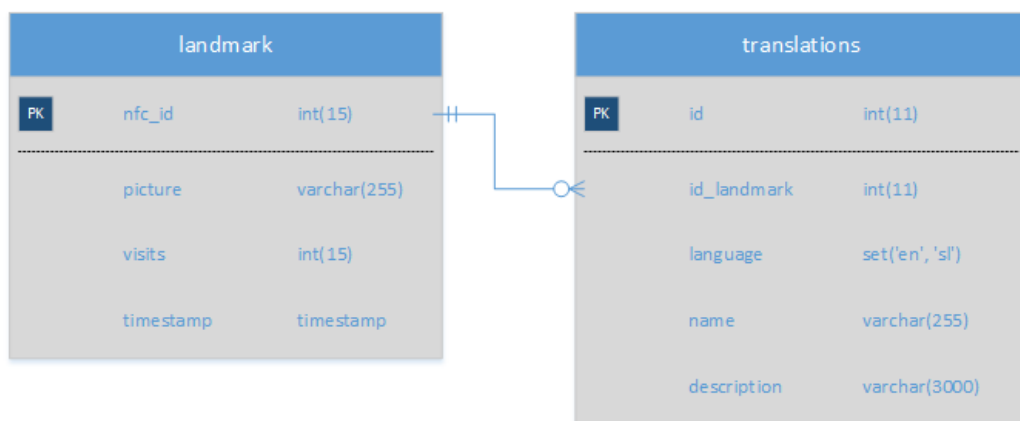
Slika 4.2: Diagram načrta komuniciranja mobilne aplikacije z internetno storitvijo REST in značkami NFC. Dopisane so posamezne tehnologije uporabljeni pri razvoju.

V nadaljevanju sta ločeno predstavljena oba omenjena dela, kjer je potrebno dodatno načrtovanje.

#### 4.1.1 Strežniški del

Na strežniškem delu se hranijo vsi ustrezni podatki znamenitosti, kateri so izpostavljeni preko spletne storitve REST. V podatkovni bazi MySQL se hranita entiteti `landmark` in `translations`, prikazani na shemi 4.3. Prva skrbi

za hrambo vseh znamenitosti, medtem ko so v drugi podatki odvisni od jezika - ime in opis znamenitosti. Vsaka znamenitost ima lahko nič ali več vnosov v entiteti `translations`, medtem ko en vnos entitete `translations` pripada samo eni znamenitosti. Entiteti sta ločeni, zaradi podpore večjezičnosti aplikacije - podprta sta slovenski in angleški jezik.



Slika 4.3: Shema podatkovne baze uporabljene na strežniškem delu sistema. Poleg imena atributa je podan še tip zapisa podatka. Vidna je tudi povezava preko tujega ključa `id_landmark`.

V nadaljevanju sledi opis atributov posameznih entitet, najprej entitete `landmark`:

- **nfc\_id**: Predstavlja enolični identifikator posamezne znamenitosti. Zapisan je na posamezno značko NFC, preko katerega mobilna aplikacija prepozna znamenitost.
- **picture**: Znakovni niz, ki predstavlja pot do slike znamenitosti na strežniku
- **visits**: Število vseh obiskov znamenitosti preko naše aplikacije
- **timestamp**: Čas, ko je bila znamenitost nazadnje spremenjena na strežniku, uporablja se v namene posodabljanja podatkov (razdelek 4.3.5).

Opis atributov entitete `translations`:

- **id**: Umetni enolični identifikator
- **id\_landmark**: Tuji ključ entitete `landmark`. Predstavlja dejansko povezavo obeh tabel, kjer ima ena znamenitost lahko nič ali več prevodov.
- **language**: Polje označuje jezik v katerem sta napisana ime in opis znamenitosti. Definiran je kot množica dveh elementov: 'en' označuje angleški jezik, 'sl' slovenskega.
- **name**: Naslov po katerem je znamenitost poznana
- **description**: Natančnejši opis znamenitosti

### 4.1.2 Del odjemalca

Na strani odjemalca je poleg lokalne baze SQLite dodatno načrtovanje potrebno še pri razvrstitvi aktivnosti in odzivanju aplikacije na določene dogodke.

#### SQLite

Lokalna podatkovna baza hrani le eno entiteto `landmarks`, prikazano na shemi 4.4. Entiteta ima podobne attribute kot strežniški entiteti, le da so dodani dodatni atributi potrebni za izvajanje aplikacije. Zato so v nadaljevanju predstavljeni samo tisti atributi, ki se razlikujejo od strežniške podatkovne baze, ostali si že predstavljeni v razdelku 4.1.1.



The image shows a screenshot of a database schema for a table named 'landmarks'. The table has the following columns and data types:

PK	_id	INTEGER
	nfc_id	INTEGER
	name	TEXT
	description	TEXT
	blob_picture	BLOB
	timestamp	TEXT
	visits	INTEGER
	my_visits	INTEGER
	visits_updated	INTEGER

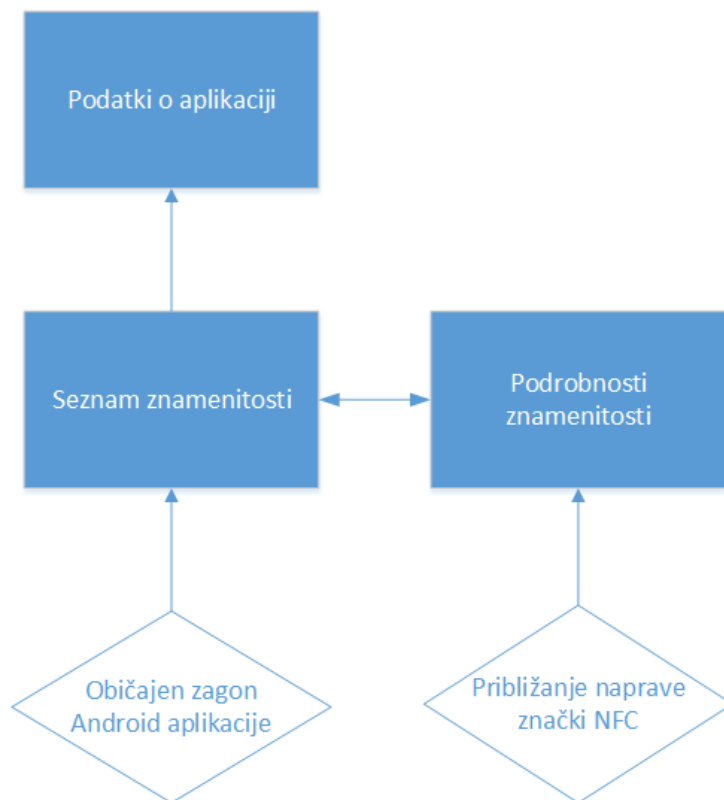
Slika 4.4: Shema podatkovne baze uporabljene na odjemalskem delu sistema.

Opis atributov lokalne entitete **landmarks**:

- **\_id**: Predstavlja umetni enolični identifikator posamezne znamenitosti. Dodan za optimiziranje SQLite poizvedb
- **picture\_path**: Ustreza atributu `picture` v strežniški entiteti **landmark**
- **blob\_picture**: Slika shranjena v podatkovni bazi v binarnem zapisu
- **my\_visits**: Število obiskov znamenitosti uporabnika aplikacije
- **visits\_updated**: Poleg polja `timestamp` se uporablja v namene posodabljanja podatkov (razdelek 4.3.5)

### Mobilna aplikacija

Aplikacija je načrtovana iz treh aktivnosti, njihove medsebojne odvisnosti in dogodki, ki jih prožijo so vidni na diagramu 4.5.



Slika 4.5: Diagram povezav in medsebojni odziv aktivnosti v aplikaciji. Modri pravokotniki predstavljajo posamezne aktivnosti, rombi pa dogodke, ki prožijo zaženejo aktivnost, ko je aplikacija ugasnjena ali v ozadju operacijskega sistema.

- **Seznam znamenitosti:** Glavna aktivnost, ki se zažene ob kliku uporabnika na ikono aplikacije. Na njej bo seznam vseh znamenitosti, kjer bo razvidno ali smo določene znamenitost že obiskali. Preko menija nastavitev bomo lahko dostopali do aktivnosti o podrobnostih aplikacije, s klikom na znamenitost pa do aktivnosti o podrobnostih znamenitosti.
- **Podrobnosti znamenitosti:** Aktivnost v kateri bodo predstavljene podrobnosti posamezne znamenitosti: ime, opis, slika, število naših in vseh obiskov znamenitosti. Do nje lahko dostopamo direktno z obiskom znamenitosti s pomočjo značke NFC ali preko glavne aktivnosti. Možen

bo prehod na glavno aktivnost.

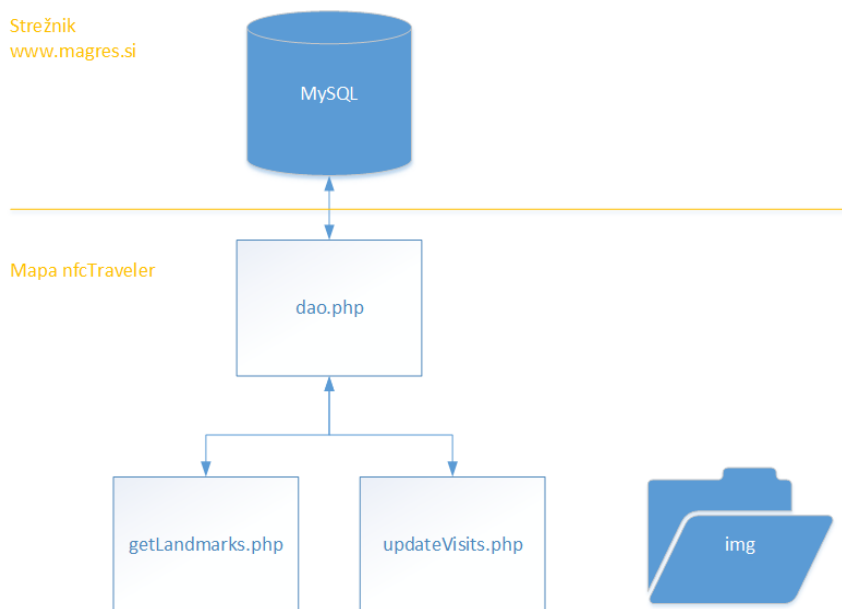
- **Podatki o aplikaciji:** Preprosta aktivnost v kateri bodo podrobnosti o aplikaciji in avtorju.

### Značke NFC

Posamezna značka bo predstavljala posamezno znamenitost. Aplikacija bo prepoznala znamenitost po enoličnem identifikatorju na znački NFC. Zapis bo enak `nfc_id` atributu v lokalni in strežniški podatkovni bazi.

## 4.2 Implementacija strežniškega dela

Pred implementacijo mobilne aplikacije bo predstavljena implementacija strežniškega dela, saj je pomemben pri razumevanju implementacije mobilne aplikacije. S pomočjo diagrama 4.6 so predstavljene glavne komponente strežniškega dela.



Slika 4.6: Diagram programskih komponent in relacij med njimi na strežniški strani. V mapi `img` najdemo ustrezne slike znamenitosti.

Na strani strežnika je uporabljena podatkovna baza MySQL, katere vsebina je opisana že pri načrtovanju, v razdelku 4.1.1. S pomočjo te baze in programskega jezika PHP so na strežniku izpostavljene tri storitve REST.

- `getLandmarks.php`: poslovna logika storitve vsebuje preprosto poizvedbo v podatkovni bazi, ki vrne vse podatke vseh znamenitosti, po imenu naraščajoče. Storitev sprejema neobvezen parameter `latest`, ki ima vrednost datuma znamenitosti z zadnjo spremembo v lokalni podatkovni bazi. V kolikor ta datum ni enak zadnjemu datumu spremembe v strežniški podatkovni bazi se pošlje odgovor z novimi podatki.

Drugi neobvezen parameter je `lang`, ki določa jezik, v katerem bodo vrnjeni podatki. Možni sta dve vrednosti: `'en'` za angleški jezik in `'sl'` za slovenski jezik. Privzeti je angleški jezik.

Podatki se prenašajo v obliki JSON.

- `updateVisits.php`: storitev skrbi za posodabljanje obiskov ustrezne znamenitosti. Mobilna aplikacija bo podala parameter `nfc_id`, ki bo določal ustrezno znamenitost. Storitev bo posodobila število vseh obiskov določene znamenitosti tako, da bo povečala trenutno vrednost za 1. Več o algoritmu za posodabljanje je napisano v razdelku 4.3.5.
- Slike znamenitosti shranjene v mapi `img`: preko storitve REST so izpostavljene tudi slike znamenitosti. Spletni naslov slike znamenitosti aplikacija pridobi preko storitve za pridobivanje vseh znamenitosti.

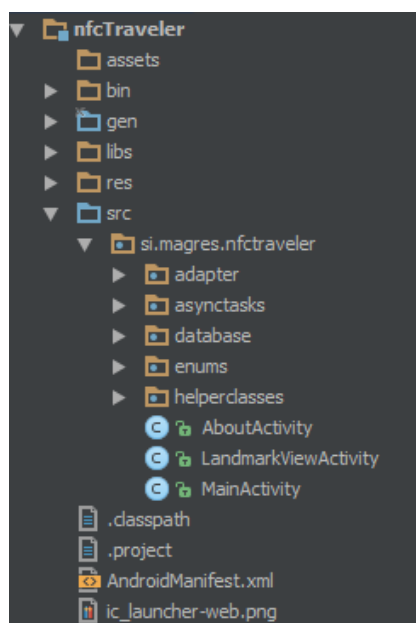
Za dostop do podatkovne baze je uporabljen za to priporočen programski vmesnik PDO. Vmesnik ponuja enake klice metod za dostop do različnih podatkovnih baz, poleg tega pa nas varuje pred napadom vrivanja ukazov SQL (ang. SQL Injection).

Vsi skupni klici omenjenih storitev, kot recimo parametri potrebni za dostop do podatkovne baze, so vključeni v skupno datoteko `dao.php`.

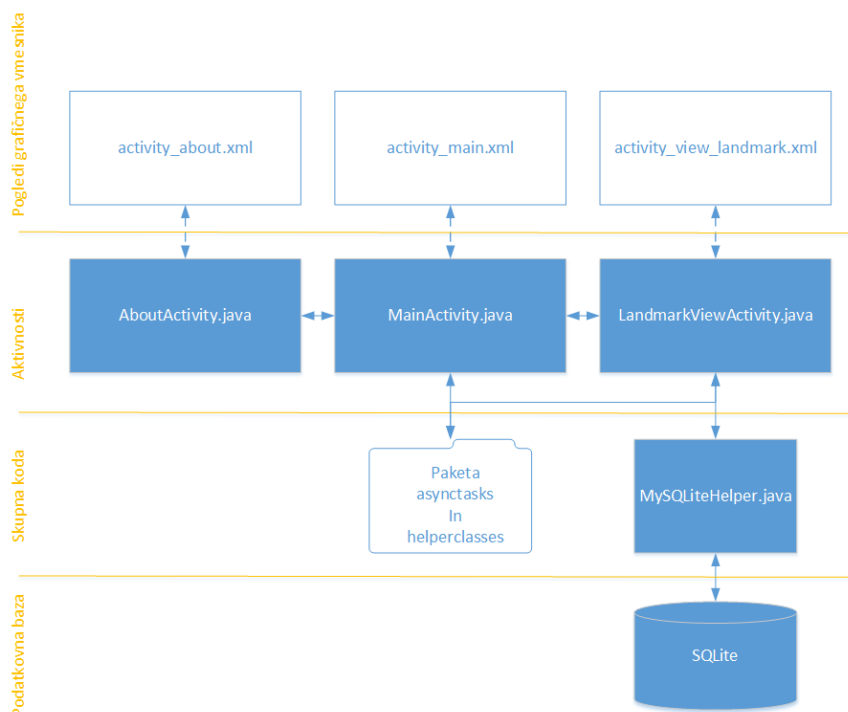
### 4.3 Implementacija aplikacije

V tem poglavju je predstavljena dejanska implementacija aplikacije načrtovane v razdelku 4.1.2. Struktura projekta iz razvojnega okolja Android Studio je predstavljena na sliki 4.7. Za samo implementacijo aplikacije sta pomembni predvsem mapi `src` in `res`. V mapi `src` je izvorna koda v javanskem jeziku smiselno razdeljena po paketih. Medtem ko so definicije postavitev, besedil uporabljenih na zaslonskih maskah, slik, animacij... v mapi `res`.

Komponentni diagram 4.8 predstavlja glavne implementacijske razrede aplikacije NFC Traveler, ki so predstavljeni v nadaljevanju.



Slika 4.7: Prikaz strukture projekta aplikacije NFC Traveler v razvojnem okolju Android Studio.



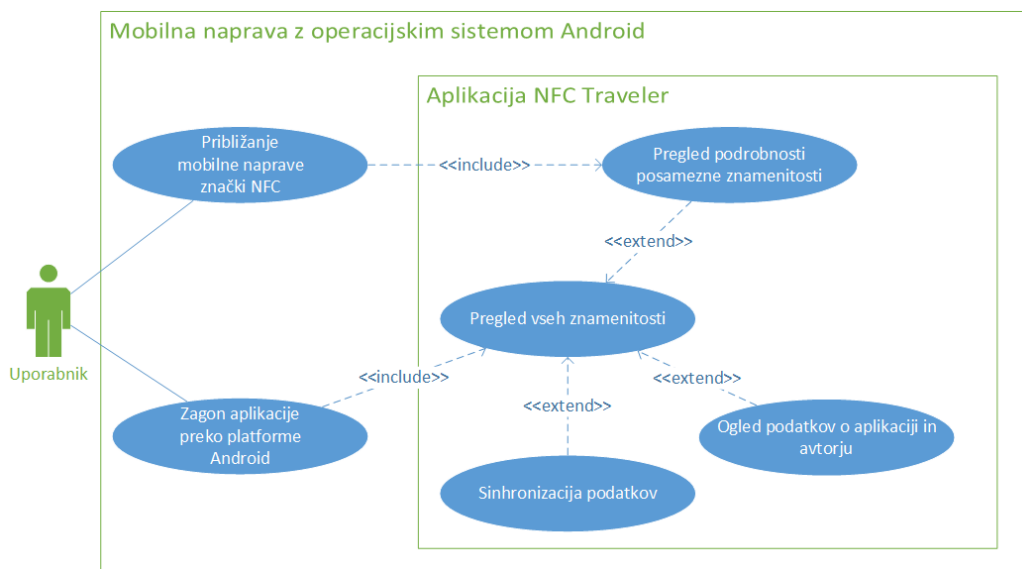
Slika 4.8: Diagram programskih komponent in relacij med njimi v aplikaciji NFC Traveler.

Za trajno shranjevanje podatkov o znamenitostih na mobilni napravi je uporabljena baza SQLite. V razredu `MySQLiteHelper.java` so vse potrebne metode za dostop in upravljanje s podatkovno bazo, kot recimo pridobivanje in shranjevanje vseh znamenitosti, pridobivanje in shranjevanje slik, urejanja posameznih atributov... Metode so klicane iz implementacijskih razredov aktivnosti in pomožnih razredov shranjenih v paketih `asyncTask` in `helperclasses`.

- **asyncTask**: vsebuje razrede, ki skrbijo, da se daljši procesi izvajajo v svoji niti, vzporedno z grafično nitjo operacijskega sistema Android. Tako nikoli ni zaustavljena grafična nit, ki skrbi za izrisovanje komponent na zaslon. V ločenih nitih se torej pridobivajo podatki iz spletnih storitev REST in podatki iz značk NFC.

- **helperclasses:** vsebuje razrede z metodami, katere so skupne več razredom. S tem ni nepotrebnega podvajanja programske kode.

Podrobnejša implementacija in delovanje aplikacije bo v nadaljevanju predstavljena s pomočjo diagrama primerov uporabe 4.9.



Slika 4.9: Diagram primerov uporabe aplikacije NFC Traveler.

### 4.3.1 Zagon aplikacije preko platforme Android

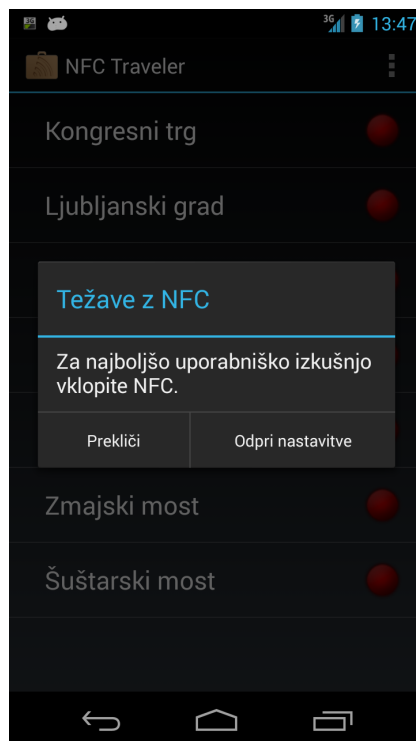
Aplikacija je zasnovana okoli strežniškega dela sistema in tehnologije NFC. Kar pomeni, da je za najboljšo uporabniško izkušnjo potrebna internetna povezava ter omogočena tehnologija NFC na napravi. Za uporabo v aplikaciji omenjenih je potrebno določiti sistemske pravice aplikaciji, kar storimo v datoteki `AndroidManifest.xml`. Dodatne pravice za uporabo tehnologije NFC niso potrebne, saj slednjo aplikacija uporablja le za branje. Potrebne so le pravice za preverjanje internetne povezave in tehnologije NFC. Točna implementacija uporabe tehnologije NFC je opisana v razdelku 4.3.4. Prav tako je zaradi uporabe tehnologije NFC aplikacija omejena na naprave z API-jem novejšim od različice 10. Realizacija pravic in minimalne ter optimalne

različice Androida v datoteki `AndroidManifest.xml` je prikazana na odseku kode 4.1.

Listing 4.1: Določanje najnižje in optimalne različice APIja naprave ter pravic za uporabo internetne povezave. Prikazan odsek kode iz datoteke `AndroidManifest.xml`.

```
1  <uses-sdk
2      android:minSdkVersion="10"
3      android:targetSdkVersion="18" />
4  <uses-permission android:name="android.permission.INTERNET"/>
5  <uses-permission android:name="android.permission.
    ACCESS_NETWORK_STATE"/>
```

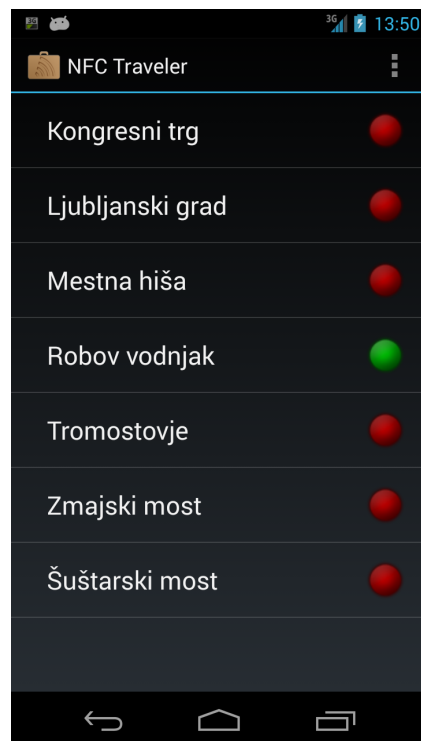
Ob zagonu aplikacije se preveri dostopnost do internetne povezave, če še ni podatkov o znamenitostih v lokalni podatkovni bazi. Uporabnika obvesti v kolikor nima omogočene tehnologije NFC, ki je priporočena za uporabo aplikacije (slika 4.10). V obeh primerih aplikacija uporabniku preko dialoga ponudi možnost dostopa do potrebnih nastavitev za vklop funkcionalnosti oziroma zaprtje dialoga.



Slika 4.10: Prikaz dialoga za obveščanje uporabnika, da nima omogočene tehnologije NFC ob zagonu aplikacije.

### 4.3.2 Pregled vseh znamenitosti

Po uspešni posodobitvi podatkov iz spletnega strežnika se prikaže seznam vseh znamenitosti, kjer s pomočjo indikatorja na desni strani vrstice lahko uporabnik vidi katere znamenitosti je že obiskal. Zelen indikator označuje že obiskane znamenitosti. Ta aktivnost je implementirana v razredu `MainActivity.java`, katere zaslonska maska je predstavljena na sliki 4.11. S klikom na posamezno znamenitost se prikaže aktivnost za prikaz podrobnosti znamenitosti.



Slika 4.11: Prikaz zaslonske maske pri pregledovanju vseh znamenitosti. Z zelenim indikatorjem so označene znamenitosti, katere smo v preteklosti že obiskali.

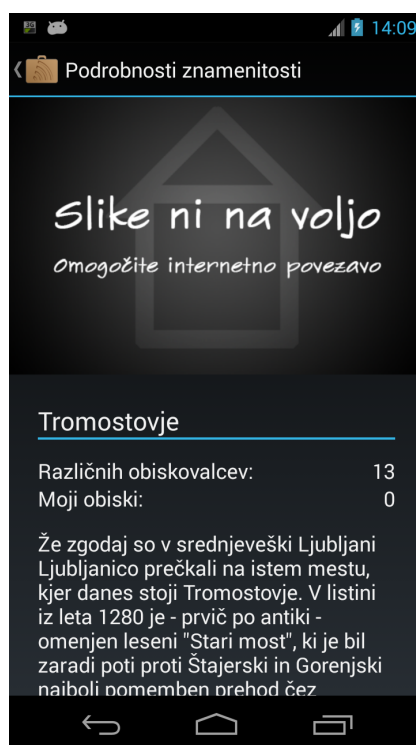
### 4.3.3 Pregled podrobnosti posamezne znamenitosti

Razred `LandmarkViewActivity.java` definira uporabniški vmesnik za pregled podrobnosti znamenitosti in vso poslovno logiko za odzivanje aplikacije na vsebino značk NFC. Slednja je predstavljena v razdelku 4.3.4, zato bo tukaj predstavljena samo ustrezna zaslonska maska.

Zaslonska maska je sestavljena iz slike znamenitosti, za katero je ustvarjen nov objekt, ki razširja objekt `ImageView` in se prilagaja širini zaslona naprave, medtem ko še vedno ohranja razmerje med višino in širino. Širina je omejena na 1000 slikovnih točk, da se na napravah z večjimi zasloni ne razširi preveč. Implementacija je v razredu `ResizableImageView.java`, ki

pripada paketu `helperclasses`. V kolikor slike v podatkovni bazi še ni in je na voljo internetna povezava se slika prenese iz strežnika ter shrani v lokalno podatkovno bazo. Če na voljo ni niti internetne povezave se prikaže privzeta slika (slika 4.12).

Pod sliko se prikaže ime znamenitosti, pod njem pa število različnih obiskovalcev in število naših obiskov znamenitosti. Za obema številoma sledi še podrobnejši opis znamenitosti.



Slika 4.12: Prikaz zaslonske maske pri pregledovanju podrobnosti znamenitosti. V tem primeru ni povezave z internetom, prav tako slike znamenitosti še ni v lokalni podatkovni bazi, zato se prikaže privzeta slika.

Prehod nazaj na aktivnost za prikaz vseh znamenitosti je omogočen s pomočjo gumba v operacijski sistem vgrajenega gumba nazaj ali preprostim potegom po zaslonu iz leve proti desni. Za bolj tekoč prehod med aktivnostma je s pomočjo animacij XML implementiran nov prehod.

### 4.3.4 Približanje mobilne naprave znački NFC

Aplikacija NFC Traveler se mora ustrezno odzvati, ko se mobilna naprava približa ustrezni znački NFC. Najprej bo predstavljena vsebina zapisana na posamezni znački, na katero se odziva aplikacija NFC Traveler in v nadaljevanju še dejanski odziv aplikacije.

#### Vsebina značke NFC

Na posamezno značko so zapisani podatki potrebni za pravilen odziv aplikacije NFC Traveler, ko omogočeno napravo NFC uporabnik približa znački. Podatki so zapisani v obliki URIja, podobnega spletnemu naslovu in formatirani po standardu NDEF (razdelek 2.1). URI je uporabljen za prepoznavanje značke s strani operacijskega sistema Android in odziva aplikacije nanjo. Kot parametra sta podana enolični identifikator `nfc_id` in ime znamenitosti. Slednji se uporablja le v primeru, da imena ne pridobimo iz strežniške podatkovne baze. V odseku kode 4.2 je primer zapisa URIja na znački NFC.

Listing 4.2: Oblika URIja zapisanega na znački NFC kjer `{nfc_id}` ustreza enoličnemu identifikatorju v podatkovni bazi `nfc_id` in `{landmark_name}` imenu znamenitosti.

```
1 <!-- Oblika zapisa: -->
2 nfc://nfcTraveler/idNum={nfc_id}&name={landmark_name}
3
4 <!-- Primer zapisa za oznacevanje Kongresnega trga -->
5 nfc://nfcTraveler/idNum=4&name=Congress Square
```

Za zapis podatkov na značke NFC v ustrezni obliki je uporabljena aplikacija TagWriter [18], katera je na voljo za brezplačen prenos preko storitve Google Play.

#### Odziv aplikacije na značko NFC

Na vsebino določeno v razdelku 4.3.4 se prvi odzove operacijski sistem Android, ki preko namenskih filtrov aplikacij (ang. `intent filter`) vsebino po-

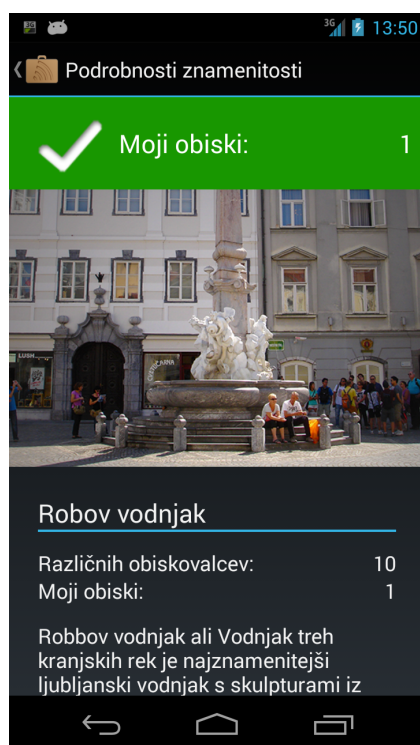
sreduje ustrezni aplikaciji oziroma aktivnosti. V aplikaciji NFC Traveler je filter nastavljen tako, da se aktivnost za ogled podrobnosti znamenitosti, `LandmarkViewActivity.java` odzove na naslove URI s korenomo `nfc://nfcTraveler` pridobljene iz značk NFC [3]. Filter je prikazan v odseku kode 4.3, kjer so pomembnejši deli dodatno komentirani.

Listing 4.3: Odsek kode odsek kode iz datoteke `AndroidManifest.xml` kjer je določena aktivnost `LandmarkViewActivity.java` za odziv na vsebino ustrezne značke NFC. Koda spada znotraj značk ki označujejo ustrezno aktivnost.

```
1  <intent-filter>
2      <!-- Odziv samo na zapis podatkov v formatu NDEF -->
3          <action android:name="android.nfc.action.NDEF_DISCOVERED"
4              />
5          <category android:name="android.intent.category.DEFAULT"
6              />
7          <!-- Odziv na sledeci URI: nfc://nfcTraveler -->
8          <data android:scheme="nfc"
9              android:host="nfcTraveler" />
10     </intent-filter>
```

Ob približanju omogočene naprave NFC značkam se torej proži aktivnost `LandmarkViewActivity.java`. Prikažejo se podrobnosti o znamenitosti iz lokalne podatkovne baze, predstavljene v razdelku 4.3.3. Hkrati se ob prvem obisku znamenitosti poveča števec vseh različnih obiskov znamenitosti v lokalni podatkovni bazi in tudi strežniški, preko storitve REST, kar je podrobneje opisano v razdelku 4.3.5. Različen obisk pomeni, da je bila znamenitost obiskana s toliko različnimi napravami. Vedno pa se poveča število obiskov uporabnika, ki se hrani samo v lokalni podatkovni bazi. Zaslonska maska ob obisku znamenitosti je prikazana na sliki 4.13.

V primeru, da zahtevane znamenitosti aplikacija ne najde v lokalni podatkovni bazi, uporabnika obvesti preko dialoga naj si posodobi podatke in ga preusmeri na aktivnost za pregled vseh znamenitosti.

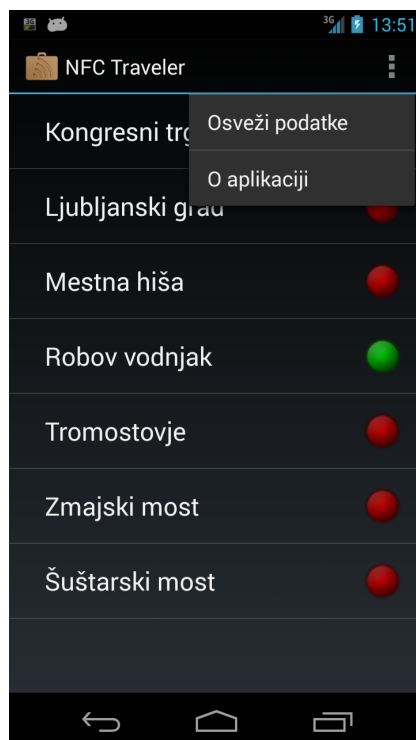


Slika 4.13: Prikaz zaslonske maske ob obisku znamenitosti s pomočjo tehnologije NFC. Pokaže se podrobnejši opis znamenitosti vključno s pripadajočo sliko in števili obiskov.

#### 4.3.5 Sinhronizacija podatkov

Za najboljšo uporabniško izkušnjo pri uporabi aplikacije je potrebno imeti vseskozi najnovejše podatke. Te podatke aplikacija dobi preko storitve REST izpostavljene svetovnemu spletu. Posodabljanje se vrši avtomatsko ob prvem zagonu aplikacije, ko je tabela znamenitosti prazna. Če je na voljo internetna povezava, se pošlje poizvedba za novejše podatke, kjer storitvi podamo parameter `latest` z vrednostjo zadnjega časovnega odtisa `timestamp` v lokalni podatkovni bazi. Podatki o vseh znamenitostih se iz strežnika prenesejo le v primeru da časovni odtis ne ustreza zadnjemu časovnemu odtisu v strežniški podatkovni bazi.

V kolikor je lokalna tabela znamenitosti prazna in ni na voljo internetne povezave nas aplikacija opozori. Posodabljanje se lahko izvede tudi ročno preko menija nastavitvev, kot je prikazano na sliki 4.14.



Slika 4.14: Prikaz zaslonske maske menija nastavitvev v glavni aktivnosti, kjer lahko s klikom na *Osveži podatke* ročno posodobimo podate.

Algoritem za posodabljanje oziroma sinhronizacijo podatkov lahko ločimo na dva dela:

### **Posodabljanje števila obiskov znamenitosti**

V tem primeru se izvede le sinhronizacija podatkov glede skupnega števila obiskov znamenitosti, ko obiščemo znamenitost. REST storitev potrebna za to sinhronizacijo je izpostavljena na naslovu: `updateVisits.php`

Poslovna logika pri tej storitvi je preprosta. Kot parameter potrebuje enolični identifikator `nfc_id`, število zapisano na NFC znački. V ozadju se

poveča števec obiskov znamenitosti za 1 enoto.

Tovrstno posodabljanje se izvaja samo v primeru obiska znamenitosti in je na voljo podatkovni dostop. V kolikor podatkovnega dostopa nimamo, si poleg števca svojih obiskov znamenitosti popravimo tudi atribut lokalne podatkovne baze `visits_updated`. Atribut ima 3 možne vrednosti:

- 0 - število skupnih obiskov ni posodobljeno niti v lokalni podatkovni bazi (znamenitosti uporabnik še ni obiskal).
- 1 - znamenitost je uporabnik že obiskal, število skupnih obiskov je popravljeno le v lokalni podatkovni bazi.
- 2 - znamenitost je uporabnik že obiskal, podatek je že posodobljen tako v lokalni, kot tudi strežniški podatkovni bazi.

Če se podatek o vseh obiskih ne posodobi tudi na strežniški podatkovni bazi, se le-ta posodobi ob naslednjem posodabljanju vseh podatkov.

### Posodabljanje vseh podatkov

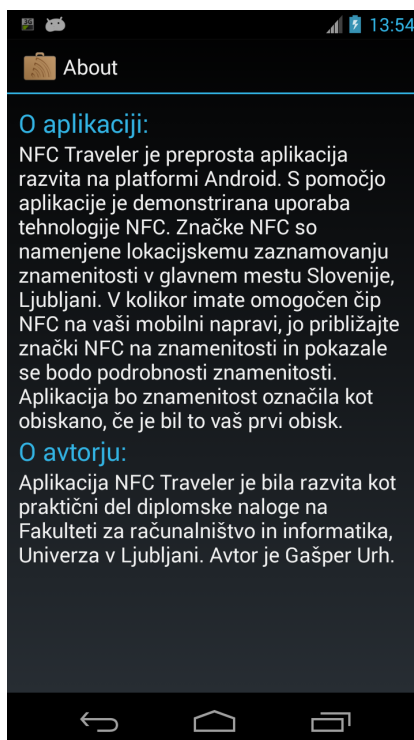
Tokrat se izvede sinhronizacija vseh podatkov strežniške podatkovne baze z lokalno podatkovno bazo. Implementacija posodabljanja se nahaja v metodi `refreshData` v razredu `DataRefresher.java`. Metoda se kliče ob ponovnem zagonu glavne aktivnosti ali na zahtevo preko menija nastavitvev, preko možnosti `Osveži podatke`.

Metoda se izvede v več korakih. V začetku se v dva ločena seznama napolnijo objekti `Landmark`, ki predstavljajo posamezno znamenitost. Eden seznam predstavlja podatke znamenitosti shranjene v lokalni podatkovni bazi, drugi pa sveže podatke znamenitosti iz strežnika. Algoritem predpostavlja, da so na strežniku vedno pravi podatki. Zato moramo izbrisati vse znamenitosti, ki so shranjene lokalno in ne (več) na strežniku. Objekte `Landmark` prepoznamo po enoličnem identifikatorju `nfc_id`. V primeru, da aktivnost obstaja, tako na strežniku, kot tudi lokalno, je potrebno posodobiti število obiskov. Potek tega dela je opisan v prejšnjem poglavju.

V naslednjem koraku iteriramo čez seznam strežniških znamenitosti in jih posodobimo lokalno. Če znamenitost lokalno že obstaja in ima datum vnosa v strežniško podatkovno bazo drugačen (polje `timestamp`), jo posodobimo na strežniško različico. V kolikor znamenitost v lokalni podatkovni bazi še ne obstaja, dodamo novo.

### 4.3.6 Podatki o aplikaciji in avtorju

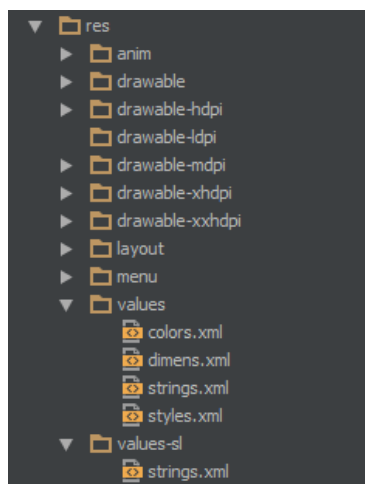
V preprosti aktivnosti `AboutActivity.java` je dodan kratek opis aplikacije in nekaj besed o avtorju. Do nje uporabnik lahko dostopa preko menija nastavitve aktivnosti za pregled znamenitosti. Zaslonska maska aktivnosti je prikazana na sliki 4.15.



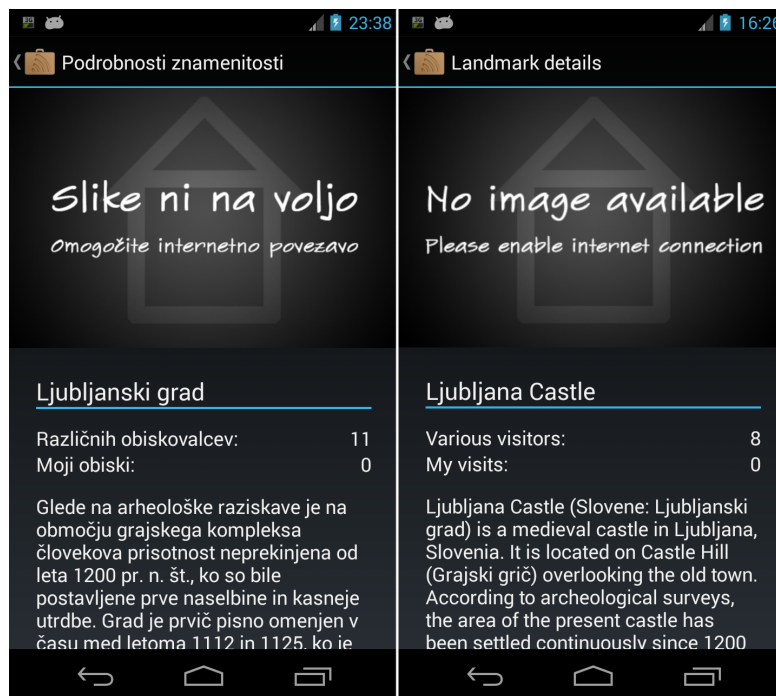
Slika 4.15: Prikaz zaslonske maske aktivnosti `AboutActivity.java`.

### 4.3.7 Večjezična podpora

NFC Traveler podpira tako slovenski kot tudi angleški jezik. Vsi nizi uporabljeni v aplikaciji so definirani v datoteki `strings.xml`, kar pomeni da moramo za prevod prevesti le to datoteko. Prevedeno datoteko moramo dodati v svojo mapo `res/values-xx`, kjer `xx` predstavlja okrajšavo za določen jezik. Operacijski sistem samodejno izbere datoteko `strings.xml` iz mape, ki ustreza jezikovnim nastavitvam operacijskega sistema. V primeru slovenščine je to mapa `values-sl`, vsebovana tudi v strukturi projekta aplikacije na sliki 4.16. Preko teh nastavitvev aplikacija poda ustrezen parameter tudi storitvi REST, da pridobi podatke v ustreznem jeziku. Če mape specifične nastavljenemu jeziku ni, se vzamejo privzete vrednosti shranjene v mapi `values`, v aplikaciji NFC Traveler so v angleškem jeziku. Na sliki 4.17 je prikazan prikaz podrobnosti znamenitosti, brez pridobljene slike, v obeh podprtih jezikih.



Slika 4.16: Struktura mape dodatnih sredstev, `res`, aplikacije NFC Traveler. V mapi `values` so shranjene privzete vrednosti, v mapi `values-sl` pa vrednosti specifične za slovenski jezik.



Slika 4.17: Primer dveh zaslonских mask, ki prikazujeta podrobnosti znamenitosti, vsaka v svojem jeziku.

### 4.3.8 Uporabljeni podatki

Podatki uporabljeni za opis znamenitosti so pridobljeni s spletne strani Wikipedia [19] z namenom demonstracije delovanja aplikacije NFC Traveler. Avtor vsega ostalega v mobilni aplikaciji - slik, nizov, itd. je avtor tega diplomskega dela.

## 4.4 Analiza delovanja aplikacije

Mobilna aplikacija NFC Traveler je namenjena širšemu krogu turistov, ki so si pripravljene na interaktivni način ogledati znamenitosti mesta. Na tem področju se je tehnologija NFC izkazala za odlično. Točno lahko označimo zeleno lokacijo, poleg tega pa omogoča izredno hiter odzivni čas. Razdalja, ki

še omogoča delovanje značk NFC uporabljenih za namen testiranja aplikacije NFC Traveler je okoli 5 centimetrov pri uporabi mobilne naprave Samsung Galaxy Nexus i9250. Na tej razdalji je odziv aplikacije hitrejši od ene sekunde, uporabnik pa ima tako na voljo podatke o obiskani znamenitosti.

Kljub vsemu, se pojavijo težave ob predstavitvi aplikacije v pravo okolje. Uporabnikom je potrebno na vidno mesto ob znamenitosti postaviti značke NFC, kar bi lahko storili s tablami, katere bi prepoznali po znaku aplikacije (slika 4.1). Prav tako nam aplikacije še ne omogoča iskanja znamenitosti s pomočjo tehnologije GPS, kar bi bila dobrodošla nadgradnja v prihodnjih izdajah za lažje iskanje znamenitosti v mestu. S širitvijo prosto dostopnih brezžičnih omrežjih v večjih mestnih jedrih in vse hitrejšimi mobilnimi omrežji se odpravijo tudi težave s podatkovno povezljivostjo in pridobivanjem podatkov iz izpostavljenih spletnih storitev. Uporabnik si tako lahko takoj pridobi oziroma posodobi podatke za delovanje aplikacije.



# Poglavje 5

## Zaključek

V diplomski nalogi je poglobljeno predstavljena tehnologija NFC skozi zgodovinski razvoj in samo delovanje. Spoznali smo, da ima tehnologija precej več prednosti kot slabosti in ji s tem lahko napovemo svetlo prihodnost. To smo potrdili tudi z analizo števila naprav z vgrajenim čipom NFC, ki so v zadnjih letih v silovitem porastu. Predstavili smo praktično uporabo tehnologije predvsem pri brezstičnem poslovanju in se spoznali s sistemi Visa payWave, MasterPass in Google Wallet. Ogledali smo si tudi ostala področja uporabe tehnologije NFC, kot recimo avtomatizacija domačih opravil.

V nadaljevanju smo se seznanili s preprosto aplikacijo NFC Traveler, s katero je prikazana uporaba tehnologije NFC. Predstavili smo vse tehnologije potrebne za razvoj aplikacije in tudi pripadajočega strežniškega dela sistema. Skozi načrtovanje, predvsem pa implementacijo smo zelo natančno spoznali delovanje mobilne aplikacije.

Zaradi uporabe spletne storitve REST na strežniškem delu je razviti sistem šibko sklopljen. Tako je zagotovljena neodvisnost od platform za morebitno nadaljnjo širitev mobilne aplikacije izven okvirjev operacijskega sistema Android. Poleg tega ima po mojem mnenju tovrstna aplikacija velik potencial in bi lahko celoten sistem korenito razširili. Prvi korak bi bil razvoj spletne aplikacije za podporo predstavljenemu mobilni aplikaciji in integracija s popularnimi družabnimi omrežji. Uporabnikom bi preko mobilne aplikacije

dovolili tudi dodajanje svojih znamenitosti na podlagi svojih uporabniških računov in tako počasi gradili skupnost uporabnikov sistema NFC Traveler.

# Literatura

- [1] “Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time,” Gartner Inc., September 2013. [Online]. Available: <http://www.gartner.com/newsroom/id/2573415>
  
- [2] V. Coskun, K. Ok, and B. Ozdeniz, *Near Field Communication (NFC): From Theory to Practice*, 1st ed. Wiley, Februar 2012.
  
- [3] —, *Professional NFC Application Development for Android*, 1st ed. Wrox, April 2013.
  
- [4] “The NFC Forum,” NFC Forum, July 2013. [Online]. Available: <http://www.nfc-forum.org/aboutus/>
  
- [5] “NFC Forum Tag Type Technical Specifications,” NFC Forum, September 2013. [Online]. Available: [http://www.nfc-forum.org/specs/spec\\_list/](http://www.nfc-forum.org/specs/spec_list/)
  
- [6] P. Jagdish Rebello, “US Wireless Carriers Partner with Big Credit Card Companies, Boosting Cell Phone NFC Market,” IHS iSuppli, August 2013. [Online]. Available: <http://www.isuppli.com/Mobile-and-Wireless-Communications/News/Pages/US-Wireless-Carriers-Partner-with-Big-Credit-Card-Companies-Boosting-Cell-Phone-NFC-Market.aspx>

- 
- [7] “Security Concerns with NFC Technology,” NearFieldCommunication.org, Avgust 2013. [Online]. Available: <http://www.nearfieldcommunication.org/nfc-security.html>
- [8] D. W. Schropfer, *The SmartPhone Wallet: Understanding the Disruption Ahead*. CreateSpace Independent Publishing Platform, December 2010.
- [9] “Visa payWave,” Visa Inc., Avgust 2013. [Online]. Available: [http://www.visaeurope.com/en/cardholders/visa\\_paywave/overview.aspx](http://www.visaeurope.com/en/cardholders/visa_paywave/overview.aspx)
- [10] “Visa and Samsung Sign Global Alliance Agreement to Accelerate Mobile (NFC) Payments,” Visa Inc., Avgust 2013. [Online]. Available: [http://www.visaeurope.com/en/newsroom/news/articles/2013/visa\\_and\\_samsung\\_nfc\\_alliance.aspx](http://www.visaeurope.com/en/newsroom/news/articles/2013/visa_and_samsung_nfc_alliance.aspx)
- [11] “Masterpass by Mastercard,” MasterCard Inc., Avgust 2013. [Online]. Available: <http://www.mastercard.ca/masterpass.html>
- [12] “Google Wallet,” Google Inc., September 2013. [Online]. Available: <http://www.google.com/wallet/>
- [13] “Platform Versions,” Android Developers, September 2013. [Online]. Available: <http://developer.android.com/about/dashboards/index.html>
- [14] M. Murphy, *Beginning Android 3*, 1st ed. Apress, Julij 2011.
- [15] “Activities,” Android Developers, Julij 2013. [Online]. Available: <http://developer.android.com/guide/components/activities.html>
- [16] “TIOBE Programming Community Index for August 2013,” TIOBE Software BV, September 2013. [Online]. Available: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [17] “Uvod v JSON,” JSON organization, Avgust 2013. [Online]. Available: <http://www.json.org/json-sl.html>

- 
- [18] “NFC TagWriter by NXP,” TagWriter, September 2013. [Online]. Available: <https://play.google.com/store/apps/details?id=com.nxp.nfc.tagwriter&hl=en>
- [19] “Wikipedia:Reusing Wikipedia content,” Wikipedia, September 2013. [Online]. Available: [http://en.wikipedia.org/wiki/Wikipedia:Reusing\\_Wikipedia\\_content](http://en.wikipedia.org/wiki/Wikipedia:Reusing_Wikipedia_content)