UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Lukić

Izvedba denarnih nakazil z mobilnimi napravami NFC

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Mira Trebar

Ljubljana, 2014

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Izvedba denarnih nakazil z mobilnimi napravami NFC

Tematika naloge:

Tehnologija NFC (Near Field Communication) omogoča enostaven brezkontakten prenos podatkov med dvema mobilnima napravama in se vse bolj uveljavlja na številnih področjih uporabe. Pametni telefon NFC lahko zamenja kreditno kartico in omogoča plačevanje storitev na PoS terminalu.

Kandidat naj preuči uporabo tehnologije NFC v pametnih telefonih, ki delujejo z operacijskim sistemom Android. Programsko ogrodje Microsoft .NET naj uporabi za izvedbo strežniške aplikacije, ki omogoča izvedbo denarnega nakazila med obstoječimi transakcijskimi računi. Prenos podatkov med dvema napravama NFC naj izvede s funkcijo Android Beam. Identifikacijski žeton naj zagotavlja preverjanje uporabnikovega računa na strežniku. Predlagana rešitev naj bo v diplomski nalogi implementirana kot mobilna aplikacija in spletna storitev tako, da se izvede prenos denarnih nakazil med testnimi transakcijskimi računi z dvema napravama NFC.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Andrej Lukić, z vpisno številko 63020099, sem avtor diplomskega dela z naslovom:

"Izvedba denarnih nakazil z mobilnimi napravami NFC"

S svojim podpisom zagotavljam, da:

1. Sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar.
2. So elektronska, oblika diplomskega dela, naslov, povzetek ter ključne besede identični s tiskano obliko diplomskega dela.
3. Soglašam z javno objavo elektronske oblike diplomskega dela v zbirki Dela FRI.

V Ljubljani, dne 15.06.2014 Podpis avtorja:

Zahvala

Za pomoč in nasvete pri izdelavi diplomske naloge se zahvaljujem mentorici doc.dr. Miri Trebar.

Za izposojeno strojno opremo (pametna telefona), sodelovanje in koristne napotke pri razvoju aplikacije se zahvaljujem sodelavcem v podjetju Halcom d.d.

Kazalo

Slovar kratic

Android SDK - Android Software Development Kit (razvojni komplet Android)
CPU - Central Processing Unit (centralna procesna enota)
ECMA – European Computer Manufacturers Association (Evropsko združenje proizvajalcev računalnikov)
EM - electromagnetic (elektromagnetni)
ETSI - European Telecommunications Standards Institute (Evropski inštitut telekomunikacijskih standardov)
HSPDA – High-Speed Downlink Packet Access (protokol mobilne telekomunikacije)
HTTP - Hypertext Transfer Protocol (protokol za prenos dokumentov preko interneta)
HTTPS - Hypertext Transfer Protocol Secure (protokol za varen prenos dokumentov preko interneta)
IEC – International Electrotechnical Commission (Organizacija za standardizacijo na področju elektronike, elektrotehnike in drugih sorodnih tehnologij)
ISO – International Organization for Standardization (Mednarodna organizacija za standardizacijo)
IMEI – International Mobile Station Equipment Identity (Mednarodna identična številka mobilne naprave)
JSON - JavaScript Object Notation (objektna notacija JavaScript)
MITM - Man In The Middle attack (napad z vmesnim členom)

MVC - Model-View-Controller (Model-pogled-krmilnik)

NDEF - NFC Data Exchange Format (standard za prenos paketa preko NFC)

NFC – Near Field Communication (komunikacija kratkega dosega)

NXP – NXP Semiconductors (podjetje, katerega ustanovitelj je podjetje Philips)

P2P - Peer to Peer (omrežje vsak z vsakim)

REST - Representational State Transfer (skupina arhitekturni omejitev)

RFID – Radio Frequency Identification (radiofrekvenčna identifikacija)

URI – Uniform Resource Identifier (povezava na določen naslov na internetu)

WI-FI – brezžična tehnologija

XML - Extensible Markup Language (razširljivi označevalni jezik)

Povzetek

Z uporabo sodobnih tehnologij je bil razvit sistem, ki s pomočjo tehnologije NFC (Near Field Communication) omogoča izvedbo denarnih nakazil med uporabniki mobilnega bančništva. Raziskana je bila tehnologija NFC, kot tudi njena uporaba na področju brezkontaktnega plačevanja ter morebitna varnostna tveganja, ki izhajajo iz strojne izvedbe tehnologije ter programske opreme za upravljanje z njo. Predlagana rešitev obsega mobilno aplikacijo napisano za operacijski sistem Android in spletni servis za nadzor in izvedbo transakcij. Predstavljen je proces načrtovanja sistema, začetna vprašanja in potek razvoja obeh aplikacij z opisi uporabljenih orodij. Izvedba sistema vključuje tehnologije, kot so Microsoftovo ogrodje .NET s pripadajočimi knjižnicami ASP.NET MVC (Model-View-Controller) za spletne aplikacije in ASP.NET WEB API za spletne servise, LINQ to SQL, integrirano razvojno okolje Visual Studio 2010, HTTP strežnik IIS, podatkovno bazo SQL Express in upravljalsko orodje za Microsoftove baze SQL Management Studio. V povezavi z razvojem mobilnih aplikacij so predstavljeni operacijski sistem Android z razvojnimi ogrodjem Android SDK (Android Software development kit), programski jezik Java in razvojno okolje Eclipse, tehnologija NFC, Android Beam in drugo. Aplikacija je bila po zaključku razvoja testirana in se bo uporabljala tudi kot prototipna demonstracijska rešitev v podjetju Halcom d.d. Njena zasnova bi lahko bila v prihodnosti vključena v razvoj mobilnih transakcij na področju gotovinskega plačevanja.

Ključne besede: NFC, NDEF, Android Beam, denarna nakazila

Abstract

Using modern tools and technologies a mobile application has been developed, which enables exchange of money for users of mobile banking. NFC (Near Field Communication) has been explored from technical perspective, as well as its use in contactless payments around the world. Safety risks have been explored, which may arise from technology itself, or its various software implementations. As a working example a system has been developed, which comprises a mobile application written for operating system Android, a web service for controlling and performing various transactions. The process of planning the system has been described thoroughly, along with the tools and technologies used. Technologies and tools described are Microsoft .NET Framework with its libraries, ASP.NET MVC and ASP.NET WEB API, LINQ to SQL, Visual Studio 2010, HTTP server IIS, SQL Express and the management tool for Microsoft databases, SQL Management Studio. In connection with developing mobile applications, the following technologies have been described: operating system Android along with Android SDK (Android Software development kit), Java and Eclipse, NFC technology and Android Beam. Application has been tested and findings and conclusions are presented at the end of this paper. The application may be used in near future as a demo application in company Halcom d.d. Its design could be included in the future development of mobile applications for contactless money transfer.

Keywords: NFC, NDEF, Android Beam, money transfer

Poglavje 1

Uvod

V času globalne razširjenosti plačilnih kartic in brezgotovinskega plačevanja se zastavlja vprašanje, kaj bo sledilo temu trendu. Vprašanja si ne zastavljajo le banke in ponudniki plačilnih kartic, temveč tudi korporacije, ki so odvisne od ponudnikov plačilnih kartic. Posrednikom plačujejo znatne provizije od vsakega izvedenega prenosa in si želijo vsaj deloma znebiti te odvisnosti. Vse to postaja zares izvedljivo šele z množičnim prodorom mobilnih naprav in dostopnostjo do interneta v večjem delu sveta. Plačevanje storitev in izdelkov z drugega konca sveta že zdaj za povprečnega prebivalca Evrope ni več nikakršna težava. Druga zgodba pa je izmenjava denarja med fizičnimi osebami. V tem segmentu še vedno gotovina nima konkurence in vprašanje je koliko časa bo še tako, oziroma kako ljudem tudi tu neopazno izmakniti kakšen odstotek.

V zadnjih letih je postalo normalno, da ima vsak pri sebi takšno ali drugačno elektronsko napravo, ki je zmožna povezovanja na internet in ima tudi neposredno možnost komuniciranja z drugimi napravami. Povsem logično je torej, da se poraja vprašanje, kdaj bo nastopil ugoden trenutek, ko bo ena izmed naprav, ki jih ljudje dnevno nosijo s seboj, prevzela vlogo denarnice in nadomestila gotovino. Vse to je že nekaj časa jasno tudi podjetju Google, ki je v navezi s peščico partnerjev, začelo uvajati v pametne mobilne telefone tehnologijo NFC (Near Field Communication). Pomanjkljivosti tehnologije

NFC postanejo prednosti ravno v svetu plačevanja in uporabi občutljivih podatkov.

Na osnovi raziskovanja različnih možnosti v razvoju plačilnih sistemov, je bila zamišljena izvedba prenosa gotovine z uporabo pametnih telefonov, ki vključujejo tehnologijo NFC. Predlagana rešitev poenostavlja delo s transakcijskimi računi z mobilno aplikacijo. Podatki transakcijskega računa so shranjeni na centralnem strežniku, ki opravlja tudi prenos denarja in skrbi za varnost in hrambo vseh podatkov. Mobilna aplikacija pa s pomočjo tehnologije NFC olajša prenos denarja med dvema računoma.

V nalogi je opisana izvedba mobilne aplikacije za operacijski sistem Android in strežniške aplikacije napisane za okolje .NET Framework. Začetna poglavja se nanašajo na opise protokolov, tehnologije NFC ter orodij in tehnologij, uporabljenih za izdelavo aplikacije. Sledi povzetek razprave o arhitekturi aplikacije, v kateri je opisan proces načrtovanja sistema ter glavni razlogi za odločitev o uporabi centralnega strežnika za hranjenje informacij in procesiranje prenosov med bančnimi računi. Po uvodni razpravi sledi podrobna predstavitev postopka izdelave spletnih servisov in mobilne aplikacije, prilagojene različnim tipom zaslonov. Rezultati testiranja aplikacije, sklepne ugotovitve in zamisli o možnih nadaljnih razširitvah aplikacije so podane v zaključku naloge.

Poglavje 2

Teoretični del

2.1 Tehnologija NFC

Komunikacija kratkega dosega (v nadaljevanju NFC) je visokofrekvenčna komunikacijska tehnologija (13.56MHz), ki omogoča izmenjavo podatkov na razdalji do 10 cm [1]. Predstavlja nadgradnjo obstoječe tehnologije RFID (Radio Frequency Identification), ki je dandanes zrela in dobro uveljavljena tehnologija. RFID omogoča čitalcu enosmerno branje podatkov s predkodirane značke ali pa pisanje na prazno značko. Zaradi cenovne ugodnosti in preprostosti je uporaba sistemov RFID priljubljena v proizvodnji, skladiščenju, sledenju produktom, plačevanju in še na številnih drugih področjih. Tehnologija NFC predstavlja logično smer nadgradnje tehnologije RFID, saj omogoča tudi dvosmerno izmenjavo podatkov in s povezovanjem z drugimi obstoječimi tehnologijami (npr. WI-FI in Bluetooth) nadgrajuje možnosti udobne in hkrati varne izmenjave podatkov med dvema napravama. V osnovi NFC temelji na treh standardih: ISO (International Organization for Standardization), ETSI (European Telecommunications Standards Institute) ter ECMA (European Computer Manufacturers Association). V letu 2003 je bila tehnologija priznana s strani ISO in IEC (International Electrotechnical Commission) kot uraden standard z oznako ISO/IEC 18092.

Leta 2004 je skupina podjetij NXP (bivši Philips Semiconductors), Sony

in Nokia ustanovila telo imenovano NFC Forum, katerega cilj je nadaljnji razvoj tehnologije ter standardov. Namen združenja je pospeševati uporabo NFC v elektronskih napravah, mobilnih telefonih in osebnih računalnikih. Promovira implementacijo in standardizacijo tehnologije NFC z zagotavljanjem združljivosti med napravami in storitvami. NFC forum šteje 150 članov (Q1, 2009) [2].

Osnova tehnologije NFC je RFID. Njeni začetki segajo v štirideseta leta prejšnjega stoletja, ko se je tehnologija prepoznave na osnovi odboja radijskih signalov začela razvijati spontano z opazovanjem pojavov pri praktični uporabi radarja. Tipična sodobna implementacija RFID tehnologije je čitalec z lastnim virom napajanja in značka (oddajnik). Čitalec oddaja elektromagnetni signal, ki se odbije od RFID značke moduliran in sprejme informacijo zapisano na njej. RFID značka je lahko pasivna (nima lastnega vira napajanja), ali aktivna (z lastnim virom napajanja). Glede na frekvenco, na katero sta usklajena oddajnik in čitalec, se deli na nizke frekvence (LF- low Frequency), 125 kHz, visoke frekvence (HF-High Frequency), 13.56MHz in ultra visoke frekvence (UHF- Ultra High Frequency), 860-960MHz. Redkeje so uporabljene tudi višje frekvence v območju mikrovalov. Različne frekvence imajo v praksi določene prednosti in slabosti in izbira prave frekvence je odvisna od mnogo dejavnikov. NFC je definiran v območju HF in deluje na frekvenci 13.56MHz. Komunikacija na tej frekvenci je predvsem omejena s kratko razdaljo med napravo in oddajnikom, ki je v praksi le nekaj cm. To pomanjkljivost so snovalci NFC obrnili sebi v prid z uporabo v situacijah, kjer je ravno fizična bližina naprave ta, ki daje večje zagotovilo za varnost transakcije. Obeti nove tehnologije so tako predvsem v domeni varnosti, izmenjave občutljivih podatkov in identifikacije. Naprava, ki podpira NFC loči tri možne načine delovanja. Prvi je tako imenovani "peer-to-peer"način, v katerem si dve napravi izmenjujeta podatke. Drugi način je branje/pisanje, ko naprava v vlogi čitalca bere ali zapisovalnika zapiše podatke na druge naprave, značke ali pametne kartice. Tretji način, emulacije kartic je usmerjen v mobilno plačevanje in je združljiv s tehnologijo brezkontaktnih plačilnih

sistemov.

2.2 Varnost tehnologije NFC

Tehnologija NFC je v osnovi zelo preprosta tehnologija radijske komunikacije. Posledica te preprostosti je žal občutljivost na več teoretičnih vrst zlorab. V nadaljevanju je predstavljenih nekaj najpogostejših zvrsti napadov, za katere se lahko s precejšnjo verjetnostjo pričakuje, da se bodo v določeni meri tudi pojavljali.

2.2.1 Prisluškovanje

Ker tehnologija NFC uporablja radijski signal, je z anteno možno prisluškovati komunikaciji med dvema napravama. Aktivni napravi je mogoče prisluškovati na daljše razdalji do 10m, medtem ko se je pasivni napravi potrebno približati na 1m. Teoretično je možno anteno skriti v bližino najverjetnejše točke prenosa podatkov in prisluškovati komunikaciji.

2.2.2 Napad s posrednikom

Napad s posrednikom je podoben napadu z vmesnim členom. Pri napadu z vmesnim členom (okrajšava: MITM (Man In The Middle attack)) se napadalec postavi med pošiljatelja in prejemnika sporočil, tako da oba mislita, da komunicirata drug z drugim, medtem pa gre njuna komunikacija v resnici skozi zlonamerni vmesni člen. Poleg prisluškovanja lahko napadalec vsebino sporočil tudi spremeni in tako aktivno poseže v komunikacijo med pošiljateljem in prejemnikom. Napad z vmesnim členom je uporaben, kadar napravi komunicirate na večji razdalji, naprimer v domačem brezžičnem omrežju, ali na svetovnem spletu. Cilj napada s posrednikom je podoben. Prestreči je potrebno komunikacijo med izvorom in ciljem ter potem prestrežena sporočila uporabiti za vdor, dostop ali drugačno zlorabo varovanih podatkov. V primeru napada s posrednikom je običajno uporabljena ravno

tehnologija komunikacije krajšega dometa, kot je ključ od avtomobila z radijskim oddajnikom, plačilna kartica z vgrajenim NFC čipom ali mobilna naprava z NFC tehnologijo.

2.2.3 Napad s spreminjanjem podatkov

Proti napadu s spreminjanjem podatkov učinkovite zaščite ni, je pa možno zaznati prisotnost napada in ukrepati z opozorilom uporabniku ali z ustavitvijo komunikacije. Naključno spremeniti podatke je precej lažje od ponarejanja podatkov, ki bodo izgledali veljavno. Pri Millerjevem kodiranju toka podatkov in 100% modulaciji je možno spremeniti le določene bite, medtem, ko je pri Manchester kodiranju z 10% modulacijo možno spreminjati vse bite [3].

2.2.4 Zaščita pred napadi

Omenjene napade je v praksi precej težje izpeljati neopazno in učinkovito, kot jih realizirati v laboratoriju ali opisati v teoriji. Razlog za to je omejitev dometa tehnologije NFC. Naprave, ki med seboj komunicirajo z NFC tehnologijo, naj bi uspešno komunicirale le v neposredni bližini, ob dotiku, oziroma skorajšnjemu dotiku. S specializiranimi antenami je sicer možno domet NFC naprave iz nekaj centimetrov povečati celo do 10 metrov, kar pa je, poleg nepraktičnosti, še vedno relativno majhen domet, ki ni primerljiv z obstoječimi komunikacijskimi tehnologijami, ki so razširjene danes. Varnosti nam domet tehnologije NFC seveda ne zagotavlja, omejuje in otežuje pa praktične možnosti in učinkovitost omenjenih napadov. Za njeno izboljšanje pri prenosu podatkov je nujna uporaba enkripcije podatkov in varnih kanalov. V kombinaciji z informiranim uporabnikom, ki napravo uporablja z določeno mero previdnosti, bi bila cena uspešnega napada verjetno dovolj visoka, da množičnih zlorab ni za pričakovati. Glede na izkušnje z obstoječimi plačilnimi sistemi lahko zaključimo, da ponudnikom storitev zadošča takšna relativna varnost.

2.3 Razširjenost tehnologije NFC

V tem trenutku še ni možno reči z gotovostjo, da je NFC tehnologija v telefonih nekaj samoumevnega. Za vgraditev NFC modula se sicer odloča vse več ponudnikov, na drugi strani pa se pojavlja množica zanimivih rešitev, ki telefonom brez tovarniške podpore tako ali drugače omogočijo vse ali le nekatere prednost tehnologije NFC. Določene rešitve se vstavijo v telefon namesto spominske kartice, druge preko izhoda za slušalke, spet tretje v obliki nalepke na hrbtni strani telefona. Na spletu je več virov, ki podrobno spremljajo podporo NFC v telefonih in redno osvežujejo seznam telefonov, ki tehnologijo podpirajo. Ena izmed takšnih spletnih strani je naprimer Nfc World [4]. Na tem mestu je nujno omeniti enega najvidnejših proizvajalcev mobilnih telefonov, podjetje Apple. Zadnjih nekaj verzij njihovega legendarnega pametnega telefona iPhone, je pred vsako uradno objavo spremljalo veliko govoric o podpori NFC tehnologiji. A po izidu zadnjih dveh verzij telefona iPhone, 5S in 5C, postaja vse bolj očitno, da podpore NFC tehnologiji v njihovem telefonu še nekaj časa ne bo. Nekateri ugibajo, da razlog tiči v nezrelosti tehnologije, ki se pozna v nekaterih nerazrešenih varnostnih vprašanjih, drugi verjamejo, da je razlog preprosto cena vgradnje v kombinaciji s premajhno popularnostjo tehnologije [5]. V vsakem primeru je jasno, da s to odločitvijo podjetja Apple, prihodnost množične uporabe tehnologije NFC v telefonih, vsaj za namene plačevanja, še vedno ni zagotovljena. Na drugi strani je dober znak, da se vsi ostali, večji proizvajalci telefonov, raje odločajo za gotovo pot in v svoje pametne telefone vgrajujejo NFC čipe.

2.4 Operacijski sistem Android in NFC

Operacijski sistem Android ima podporo NFC modulom že vgrajeno. V nadaljevanju sledi opis možnih načinov uporabe uradnih Android knjižnic za prejemanje in pošiljanje NFC podatkov v obliki NDEF (NFC Data Exchange Format) paketov. Obstajata dve možni situaciji, kjer se pojavljajo NDEF podatki na Androidu:

1. Branje NDEF podatkov z NFC značke (NFC tag).

2. Prenos NDEF sporočil z ene naprave na drugo z uporabo Android storitve Android Beam.

Ko naprava prebere NFC značko, je potrebno vsebino prebranih podatkov najprej analizirati in ugotoviti, kateri uporabniški aplikaciji posredovati prejete podatke. Podatki so namreč lahko zelo raznovrstni, od preproste interne povezave na določeno stran, do poljubnih tekstovnih in drugačnih vrst podatkov. Za analizo podatkov skrbi sistem za analizo in razporejanje prebranih značk (ang. tag dispatch system). Le-ta je vgrajen že v operacijski sistem Android in njegova naloga je ugotoviti, katera izmed aplikacij, ki so nameščene na napravi, bo najbolj primerna za odpiranje in nadaljno obdelavo podatkov na NFC znački. Vsaka uporabniška aplikacija se ima pravico in možnost registrirati kot primerna za obdelavo vseh ali le določene vrste podatkov na NFC značkah. Podrobneje je sistem za analizo in razporejanje NFC značk opisan v nadaljevanju.

2.4.1 NDEF format

NDEF je okrajšava za NFC Data Exchange Format in predstavlja format in nabor pravil za zapis podatkov na NFC značkah. NDEF sporočilo sestoji iz enega ali več NDEF zapisov. Število posameznih NDEF zapisov zavisi od vrste aplikacije in tipa značke.Če je uporabljeno NDEF sporočilo, ki hrani povezavo do določene spletne strani, je potreben le en NDEF zapis. Sporočilo bo prenešeno v binarni obliki, kjer je v formatu določen pomen vsakega zloga [6].

2.4.2 Android Beam

Storitev Android Beam omogoča, da mobilna naprava prenese NDEF sporočilo na drugo napravo, tako da uporabnika stakneta napravi skupaj. Tak prenos je enostavnejši od ostalih brezžičnih tehnologij (WI-FI, Bluetooth), saj ni

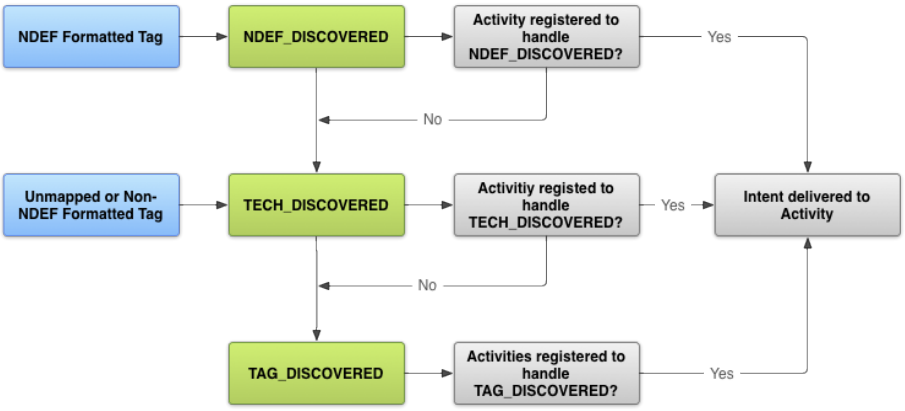
potrebno uparjanje in iskanje naprave. Povezava se vzpostavi avtomatsko, ko se napravi približata druga drugi na primerno razdaljo. Funkcionalnost Android Beam je preko Android knjižnic za uporabo na voljo vsaki aplikaciji [7].

2.4.3 Sistem za razporejanje prebranih značk

Sistem za razporejanje prebranih značk je aktiven, kadar naprava ni v zaklenjenem stanju. Ko naprava zazna značko, je naloga sistema, da ugotovi, katera aplikacija bo najbolj primerna za obdelavo podatkov na znački, ne da bi bilo potrebno posredovanje uporabnika naprave. Ugotavljanje tipa podatkov poteka tako, da se prebere vsebina prvega zapisa v NDEF sporočilu (NDEF sporočilo lahko vsebuje poljubno število zapisov). V prvem zapisu sporočila je podan tudi tip podatkov. Aplikacije morajo težiti k temu, da čim natančneje definirajo tip značk, ki jih želijo obdelovati. Sistem deluje v treh korakih:

1. Branje NFC značke in ugotavljanje tipa zapisanih podatkov.
2. Enkapsulacija vsebine značke v razred za nadaljno obdelavo v aplikaciji.
3. Posredovanje informacij izbrani aplikaciji za nadaljno obdelavo.

Ko je tip podatkov poznan, poskuša sistem zagnati aplikacijo, ki je registrirana za obdelavo tega tipa podatkov. Če takšne aplikacije ni, poskuša sistem najti aplikacijo, ki je registrirana za obdelavo širše kategorije podatkov in na ta način sestopa do najbolj splošne možne aplikacije, ki bere vse NFC značke. Da bi lahko naša aplikacija obdelovala določen tip značk, je potrebno to deklarirati v konfiguracijski datoteki Android.xml. Za primer deklaracije se lahko vzame aplikacijo, ki uporablja svoj tip značke in ga želi tudi obdelovati. Deklaracijo se zapiše v aktivnost, ki naj se odpre, ko operacijski sistem zazna ustrezno značko. Slika 2.1 prikazuje potek analize NDEF paketa.



Slika 2.1: Analiza prebrane NFC značke [8]

Poglavje 3

Izvedba denarnega nakazila

3.1 Uvodna razprava

Arhitektura aplikacije je bila na začetku predmet daljše razprave, saj je bil prvotni cilj izdelati elektronski ekvivalent običajni gotovini. Ker je posamezna enota gotovinskega denarja fizično prosto izmenljiva na enostaven način, brez zahtevane prisotnosti kakršnega koli nadzora, smo se poigravali z idejo, da bi strežniško aplikacijo izločili in odgovornost izvajanja transakcij v celoti prepustili mobilni napravi sami. Ker bi na ta način lahko deloval tudi brez prisotnosti omrežne povezave in brez nadzora strežniške aplikacije, bi bile prednosti velike. Transakcije bi potekale hitro, enostavneje in zanesljivo. Morda najvažneje za samo rešitev - potekale bi brezplačno in neodvisno od mobilnega operaterja, denar bi lahko brez težav izmenjevali v tujini, v načinu gostovanja pri tujem mobilnem operaterju. Kljub privlačnosti ideje same smo na koncu izbrali alternativno pot. Glavni razlog za to je bil širok spekter možnih zlorab takega sistema in mobilne tehnologije, ki so še v fazi dozorevanja.

Strežniške tehnologije (operacijski sistemi *nix, MS Windows, aplikacijska okolja Java, ogrodje .NET, spletni strežniki Apache, IIS, komunikacijski protokoli HTTP (Hypertext Transfer Protocol), HTTPS (Hypertext Transfer Protocol Secure) so danes večinoma dozorele, zato se je njihov razvoj

delno upočasnil. Stopnja zaupanja v te tehnologije je relativno velika, ker dandanes poganjajo številne občutljive sistem po vsem svetu. Ob pravilni uporabi zagovljajo varnost, skalabilnost in robustnost. Mobilne tehnologije so na drugi strani v fazi hitrega vzpona in živahnega razvoja tako na strani strojne opreme, kot na strani programske opreme. Če pogledamo operacijska sistema Android in njegovega vzornika in konkurenta, iOs, sta oba podvržena izjemno hitrim nadgradnjam, ki se pogosto dotikajo vitalnih delov operacijskega sistema. Operacijski sistemi mobilnih naprav so si zadali ambiciozen cilj široke podpore raznovrstnim strojnim konfiguracijam, ki se lahko med seboj precej razlikujejo. Če so si mobilni telefoni in tablice še dokaj podobni, pa sta odmeven poskus pametnih očal in pametne ure že korak k večji raznovrstnosti naprav podprtih s strani mobilnih operacijskih sistemov. Spremembe v mobilnih operacijskih sistemih poskušajo na eni strani slediti izboljševanju in dodajanju nove strojne opreme in na drugi strani uporabniku kljub vse večji kompleksnosti naprave oz. množice naprav, ponuditi še bolj intuitiven in preprost uporabniški vmesnik. Ocenili smo, da bi bilo tvegano v tako nestabilnem okolju zagotoviti varnost hranjenja in prenašanja denarnih sredstev brez nadzora. Iz naštetih razlogov smo se odločili za klasični pristop vsevednega strežnika in tankega odjemalca. Tak sistem prelaga večino odgovornosti na strežnik in seveda zahteva prisotnost internetne povezave in ustrezne strežniške aplikacije. Mobilne naprave v medsebojni komunikaciji zgolj prenašajo minimalne informacije potrebne za izvajanje transakcij na strežniku. V našem primeru so to unikatni identifikacijski žetoni, preko katerih strežnik prepozna uporabnika v svojem sistemu.

3.2 Arhitektura aplikacije

Sistem za prenos denarnih nakazil sestoji iz odjemalčeve aplikacije (ang. client application), ki teče na mobilni napravi ter strežniške aplikacije (ang. server application), ki teče na strežniku in skrbi za varno izvajanje transakcij med bančnimi računi. Mobilna aplikacija ima vlogo odjemalca in vse

podatke v realnem času pridobi od strežnika. Prav tako vsako operacijo v izvajanje pošlje na strežnik. Edina informacija, ki jo mobilna naprava fizično hrani v spominu, je identifikacijska številka uporabnika, ki enolično določa bančni račun v bazi strežnika. V primeru prenosa denarja, prenos podatkov za izvedbo transakcije med obema mobilnima napravama poteka preko tehnologije Android Beam, sam zahtevek za prenos pa mobilna naprava na strežnik prenese preko protokola HTTP, torej preko svetovnega spleta. Mobilna naprava mora biti torej na nek način povezana v omrežje. To je lahko 2G, 3G, 4G ali brezžičen dostop (WI-FI).

Mobilna naprava je v našem konkretnem primeru lahko katerikoli pametni telefon z operacijskim sistemom Android, od verzije 4.1 (Jelly Bean) dalje. Verzija je predvsem določena s podporo funkcionalnosti Android Beam, ki omogoča enostaven in hiter prenos podatkov preko tehnologije NFC in Bluetooth in nekaj dodatnih podpornih funkcij. Strežniška aplikacija je napisana v ogrodju Microsoft .Net Framework, verzije 4.5. Tehnologija spletnega servisa je Asp.Net Web Api, jezik pa C#. Strežniška aplikacija lahko teče na kateremkoli strežniku, na katerega je mogoče namestiti .Net ogrodje. Zaenkrat so to le Microsoftovi operacijski sistemi, saj verzija .Net ogrodja na operacijskem sistemu Linux v tem trenutku zaostaja s podporo. HTTP strežnik je Microsoftov IIS verzije 7, podatkovna baza pa brezplačna različica Microsoftove podatkovne baze SQL Express, kar pomeni, da mora biti tudi ta nameščen na operacijskem sistemu. Načeloma bi vrsto podatkovne baze lahko enostavno nadomestili s kakšno odprtokodno podatkovno bazo, saj so uporabljene le standardne funkcionalnosti, ki so podprte tudi v MySQL bazi.

3.3 Podatkovni model

Podatkovni model mora hraniti potrebne podatke o bančnih računih uporabnikov elektronskega računa, kot tudi o zgodovini prenosov med računi. V vsakem trenutku mora biti iz vseh odlivov in prilivov jasno razvidno trenutno stanje na računu. Uporabiti je torej potrebno vsaj dve entiteti. Prva entiteta,

poimenovana Računi, predstavlja bančni račun posameznega uporabnika z vsemi pripadajočimi informacijami. Druga entiteta, poimenovana Prenosi, pa predstavlja zgodovino prenosov med posameznimi bančnimi računi. Entiteta Račun potrebuje atribut identifikacijsko zaporedje znakov, ki vsakega uporabnika v sistemu enolično identificira. Zaradi udobnejšega sklicevanja na ta atribut je naziv skrajšan v ID računa uporabnika. Potrebne so še informacije o stanju na računu in še nekaj manj pomembnih atributov; ti so namenjeni boljši uporabniški izkušnji in lepšemu delovanju aplikacije. Atributi entitete Računi so po vrsti:

1. **ID računa** - enolično identificira nosilca računa v sistemu.
2. **Stanje** - znesek stanja v privzeti valuti.
3. **PIN številka** - dodatna zaščita za mobilno aplikacijo.
4. **IBAN** - standardizirana številka transakcijskega računa.
5. **Ime** - ime nosilca računa.
6. **Priimek** - priimek nosilca računa.

Entiteta Prenosi predstavlja zgodovino prenosov med računi posameznih uporabnikov. V to kategorijo spada teoretično tudi nalaganje denarnih sredstev na račun, če jih sistem podpira. V tem primeru je pošiljatelj kar ponudnik storitve, oziroma je polje prazno. Uporabljeni atributi v entiteti so:

1. **ID računa pošiljatelja** - identificira pošiljatelja.
2. **ID računa prejemnika** - identificira prejemnika.
3. **Datum prenosa** - trenutek, ko je bil prenos izveden.
4. **Znesek** - vrednost prenosa v privzeti valuti.
5. **Opis** - deskriptivni opis za jasnejši pregled v aplikaciji.

Račun vsakega uporabnika lahko nastopa tako v vlogi pošiljatelja v več prenosih, kot v vlogi prejemnika v več prenosih. Zato je potrebno definirati med entitetama Računi in Prenosi še dve relaciji ena proti več. Podatkovna baza je relacijska. Uporabljena je brezplačna podatkovna baza proizvajalca Microsoft, SQL Express. Za delo pa je na voljo orodje SQL Managment Studio, ki omogoča podporo vsem potrebnim administrativnim in razvijalskim nalogam.

Za našo konkretno implementacijo, je v podatkovni bazi Microsoft SQL Server ustvarjena tabela Računi, ki vsebuje podatke o računih uporabnikov in sestoji iz stolpcev(v oklepaju je navedeni tip polja v podatkovni bazi Sql Express in oznaka "not null", če polje ne sme biti prazno):

1. **Id** (integer, identity, not null) - ID računa in primarni ključ tabele.
2. **Guid** (uniqueidentifier, not null) - 32 znakov dolgo zaporedje naključnih znakov.
3. **Pin** (varchar(10), not null) - štirimestna varnostna številka, s katero uporabnik dostopa do mobilne aplikacije.
4. **Ime** (nvarchar(50), not null) - ime nosilca računa.
5. **Priimek** (nvarchar(50), not null) - priimek nosilca računa.
6. **Stanje** (money(50), not null) - denarno stanje računa.
7. **Iban** (nvarchar(50), not null) - številka transakcijskega računa uporabnika.

Ustvarjena je še tabela Prenosi, ki sestoji iz naslednjih stolpcev:

1. **Id** (integer, identity, not null) - ID je primarni ključ tabele.
2. **IdPošiljatelja** (int, not null) - je zunanji ključ, ki identificira račun uporabnika v tabeli Računi.

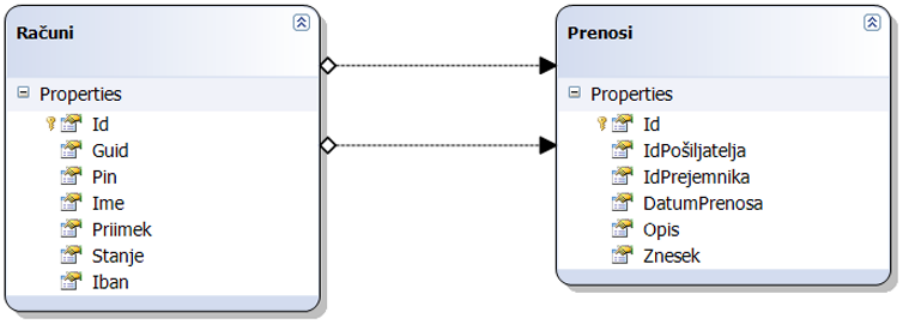
3. **IdPrejemnika** (int, null) - je zunanji ključ, ki identificira račun uporabnika v tabeli Računi.

4. **DatumPrenosa** (datetime2(7), not null).

5. **Opis** (nvarchar(50), not null).

6. **Znesek** (money, not null) - prenešeni znesek.

Slika 3.3 prikazuje tabeli spletnega servisa.



Slika 3.1: Tabele podatkovnega modela

3.4 Spletni servis

Najpomembnejšo vlogo v celotnem sistemu ima centralni strežnik, ki nosi odgovornost za hranjenje podatkov o posameznih računih, zgodovini prenosov in opravlja same prenose med računi. Naštete naloge opravlja spletni servis s podatkovno bazo in je javno dostopen na internetu preko HTTP protokola. Za implementacijo spletnega servisa je izbran arhitekturni stil REST (REpresentational State Transfer), ker je združljiv s protokolom HTTP in temu primerno zelo enostaven. Spletni servis mora podpirati vsaj tri osnovne naloge, ki bodo opisane v nadaljevanju.

3.4.1 Uporabljena programska oprema

Microsoft .NET ogrodje

Programsko ogrodje .Net Framework podjetja Microsoft je tehnologija, ki olajša izdelavo aplikacij in XML (Extensible Markup Language) spletnih servisov. Glavni cilji ogrodja .Net Framework so objektno orientirano programiranje, minimiziranje konfliktov pri postavitvi aplikacije, zagotovitev varnega izvajanja programske kode, konsistentna razvijalska izkušnja ne glede na tip aplikacije in sledenje standardom, da bi bila napisana programska koda čim bolj združljiva med seboj. Ogrodje sestoji iz jedra Common Language Runtime (krajše CLR) in .NET Framework knjižnice. CLR v obliki navideznega stroja opravlja s programsko kodo v času izvajanja in zagotavlja osnovne funkcije, kot so: delo s spominom, delo z nitmi, dela z napakami in izjemami [9].

REST (Representational State Transfer) arhitektura

Kratica REST (Representational State Transfer) označuje arhitekturni stil šibko sklopljenih sistemov. Sestavljen je iz odjemalcev in strežnikov. Odjemalec preko spletnega naslova (URI) odda zahtevo, strežnik zahtevo obdela in odjemalcu vrne odgovor. Rezultat je lahko v različnih oblikah, najpogosteje uporabljeni obliki sta JSON (JavaScript Object Notation) in XML. Veljajo naslednje omejitve:

1. Odjemalec in strežnik sta ločena. Na tak način se zagotovi neodvisnost uporabniškega vmesnika od razvoja strežnika.

2. Ni stanja, saj vsaka zahteva vsebuje vse potrebne podatke za obdelavo, zato strežnik med prenosi ne shranjuje nobenih vmesnih podatkov.

3. Odjemalci lahko shranjujejo rezultate obdelanih zahtevkov v predpomnilnik, zato morajo biti odgovori s strežnika jasno definirani tudi glede shranjevanja v predpomnilnik.

4. Enotni vmesnik med komponentami zagotovi preglednejšo arhitekturo, izgubi pa se nekoliko pri učinkovitosti, saj se informacije prenašajo v enaki obliki, ne glede na to, ali je to za aplikacijo optimalno ali ne.

5. Večnivojska zasnova aplikacije.

ASP.NET MVC

ASP.NET MVC je zelo razširjena tehnologija za razvoj spletnih strani in je nadgradnja starejše tehnologije ASP.NET. Prednosti so predvsem v uporabi vzorca Model-View-Controller (MVC). Ta razvijalski vzorec nam olajša delitev aplikacije na tri komponente: model (nastopa kot podatkovni vir in vsebuje logiko za delo s podatki aplikacije, naprimer pridobivanje in shranjevanje podatkov v podatkovno bazo), pogled (prikazuje podatke modela in predstavlja uporabniški vmesnik) in kontroler (nadzoruje in vodi tok dogodkov aplikacije, koordinira model s pogledom). Omogoča tudi večji nadzor nad usmerjanjem akcij in obnašanjem aplikacije (ukinjena je funkcionalnost view state, ki se uporablja v tehnologiji ASP.NET) [10].

ASP.NET Web API

ASP.NET Web API je ogrodje, ki poenostavlja razvoj HTTP servisov namenjenih široki paleti odjemalcev, vključno z spletnimi brskalniki in mobilnimi napravami. To je idealna platforma za razvoj REST (Representational State Transfer) aplikacij na ogrodju .NET Framework, posebej ugodna za razvijalce z izkušnjami v tehnologiji ASP.NET MVC [11].

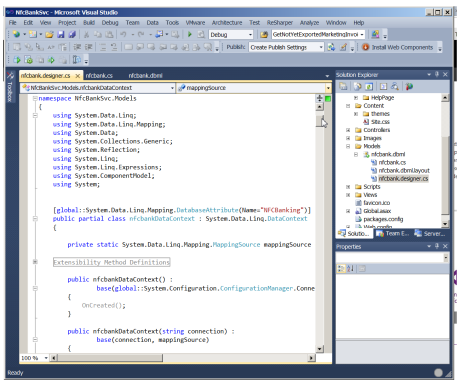
LINQ to SQL

LINQ to SQL je komponenta ogrodja .NET Framework, ki omogoča objektno delo s podatki relacijskih podatkovnih baz. Podatkovni model relacijske podatkovne baze se preslika v objektni model, izražen v programskem jeziku razvijalca. Ko aplikacija teče, LINQ to SQL preslika poizvedbe iz programskega jezika v SQL in pošlje podatkovni bazi. Rezultat podatkovne baze nato

preslika nazaj v objekte s katerimi dela aplikacija [12].

Visual Studio 2010

Aplikacija Visual Studio (slika 3.2) podjetja Microsoft je vsestranska zbirka orodij in servisov, namenjenih razvoju široki paleti različnih aplikacij; podpira razvoj namiznih aplikacij, spletnih servisov, spletnih strani, oblačnih aplikacij, ... Visual Studio vsebuje tudi orodja za delo v skupini, v kateri so posamezniki lahko tudi na drugačnih platformah [13].



Slika 3.2: Microsoft Visual Studio 2010

Microsoft SQL Express

Microsoft SQL Express je brezplačna programska oprema podatkovne baze, ki je združljiva s katerimkoli ogrodjem za izdelavo spletnih aplikacij. Prenos je brezplačen in neomejen, prav tako razmnoževanje. Namenjena je manjšim aplikacijam in vgrajenim sistemom ter vsebuje veliko funkcionalnosti plačljive, polne verzije podatkovne baze. Omejitve so tehnične narave; največja velikost baze na disku je 10GB, uporablja lahko največ eno CPU (Central Processing Unit) in uporablja največ 1GB spomina [14].

3.4.2 Implementacija

Programsko rešitev sestavlja en projekt, ki vsebuje dva kontrolerja in LINQ to SQL avtomatsko generirane objekte, ki predstavljajo podatkovni model. Oba kontrolerja vračata rezultat v obliki XML zapisa. Prvi kontroler je namenjen vračanju podrobnosti o posameznem bančnem računu (vpogled v stanje računa) in opravljanju prenosa z enega računa na drug račun. Drugi kontroler vrača zgodovino vseh odlivov in prilivov podanega računa.

Vpogled v stanje računa

Za vpogled v stanje računa spletni servis potrebuje le en vhodni podatek in sicer ID računa, ki enolično določa uporabniški račun v sistemu. Če račun v bazi obstaja, spletni servis dostopi do podaktov o stanju računa in informacijo vrne v obliki XML odgovora.

Vpogled v zgodovino prenosov računa

Za vpogled v zgodovino prenosov med računi, spletni servis potrebuje en vhodni podatek, ID računa. Če račun v bazi obstaja, spletni servis dostopi do seznama vseh prenosov, v katerih podani račun nastopa v vlogi pošiljatelja in nato še seznama vseh prenosov, v katerih navedeni račun nastopa v vlogi prejemnika. Oba seznama nato združi v enoten seznam v padajočem vrstnem redu po datumu prenosa. Rezultat nato vrne v XML obliki.

Zahtevek za prenos med računoma

Za prenos poljubnega zneska z enega računa uporabnika na drug račun mora spletni servis prejeti ID obeh računov in željeni znesek. Izpolnjen mora biti pogoj, da oba računa obstajata ter da ima pošiljatelj na svojem bančnem računu stanje, ki je večje ali enako željenemu znesku prenosa. Če so pogoji izpolnjeni, se prenos izvede, kar pomeni, da se stanje na računih obeh uporabnikov ustrezno spremeni, pošiljatelju se znesek odšteje, prejemniku prišteje, v tabelo, ki hrani zgodovino prenosov med računi pa se zapiše nova

vrstica o izvedenem prenosu. Če prenos slučajno ne uspe, spletni servis vrne napako in stanje na obeh računih ostane nedotaknjeno.

3.4.3 Strojna oprema

Testni strežnik

Za namestitev spletnih servisev je bil uporabljen osebni računalnik, tipa HP dc7100 (slika 3.3), z nameščenim operacijskim sistemom Microsoft Windows 7 Professional in programskim okoljem Microsoft .Net Framework 4.5, podatkovno bazo Microsoft SQL Express in spletnim strežnikom IIS 7. Strojne specifikacije osebnega računalnika:

1. Procesor Intel Pentium 4, 3GHz.
2. 3GB RAM vrste DDR DRAM PC3200 (400MHz) ne-ECC.
3. 500GB SATA trdi disk.
4. Integrirana 1Gbit mrežna kartica.

Razvojni računalnik

Za razvoj mobilne aplikacije in spletnih servisov je bil uporabljen prenosni računalnik, tipa Lenovo W510 (slika 3.3), z naslednjimi specifikacijami:

1. Procesor Intel Core i7-920XM, 2GHz, 8MB cache.
2. 12GB RAM vrste DDR3.
3. 500GB SATA trdi disk.
4. Integrirana 1Gbit mrežna kartica.

Mobilna telefona s podporo NFC tehnologiji

Za testiranje mobilne aplikacije sta bila uporabljena dva mobilna telefona proizvajalca Samsung, vrste Galaxy S3 mini (slika 3.3). Specifikacija telefonov, ki so pomembne za delovanje aplikacije:

1. Procesor 1GHz dual-core Cortex-A9.
2. 1GB RAM.
3. NFC modul.
4. Integrirana 1Gbit mrežna kartica.

Na obeh telefonih je bil nameščen operacijski sistem Android OS, verzije v4.1 (Jelly Bean).

Tablični računalnik Samsung Nexus 10

Za testiranje aplikacije na tabličnem računalniku je bil uporabljen zmogljiv tablični računalnik Samsung Nexus 10.

1. Dual-core 1.7 GHz Cortex-A15.
2. 2 GB RAM.
3. NFC modul.
4. Integrirana 1Gbit mrežna kartica.
5. Zaslon, ločljivosti 2560 x 1600 pixels, 10.1 inčev (299 ppi gostota pikslov).

Na tabličnem računalniku je bil nameščen operacijski sistem Android OS, verzije v4.3 (Jelly Bean).

Slika 3.3: Strojna oprema

3.5 Mobilna aplikacija

Vlogo odjemalca predstavlja mobilna aplikacija, ki se izvaja na napravi, kot je pametni telefon. Izpolnjena morata biti dva pogoja:

1. Omogočena NFC tehnologija in
2. možnost povezave na internet.

Mobilna aplikacija služi kot uporabniški vmesnik za delo s spletnim servisom. Uporabniku bo omogočala vpogled v stanje računa, vpogled v zgodovino prenosov z računa in na račun ter sprožitev zahtevka za prejem poljubnega zneska z drugega računa.

3.5.1 Uporabljena programska oprema

Java

Java je objektno usmerjeni, prenosljivi programski jezik, ki ga je razvil James Gosling s sodelavci v podjetju Sun Microsystems. Projekt, ki se je v začetku (leta 1991) imenoval Oak (hrast), je bil razvit kot zamenjava za C++. Jave se ne sme zamenjevati z jezikom JavaScript, ki ima podobno ime, ter podobno, C-jevsko skladnjo. Različica Java 1.0 je bila objavljena leta 1996, zadnja različica je 6.0 (avgust 2007). Javo vzdržuje in posodablja Oracle - Sun

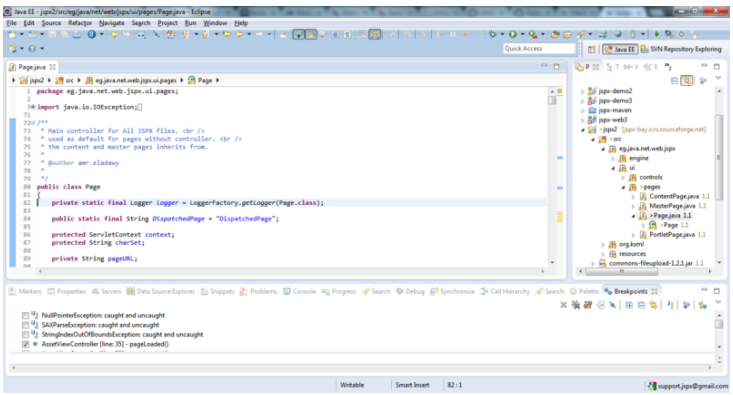
Microsystems. Tolmač za programski jezik Java je vgrajen v večino spletnih brskalnikov, s tem se javanski programčki (applet) lahko izvajajo kot del HTML dokumenta [15]. Obstajajo 3 različice:

1. **J2SE** - standardna različica Java za osebne računalnike.
2. **J2ME** - različica Java za naprave (mobiteli, pametni televizorji, ...).
3. **J2EE** - poslovna različica Java.

Java se uporablja tudi za programiranje aplikacij na mobilnih telefonih (J2ME) in pametnih telefonih z operacijskim sistemom Android.

Eclipse

Eclipse (slika 3.4) je brezplačno, odprtokodno, integrirano okolje (IDE) za razvoj programskih aplikacij v Javi in ostalih podprtih programskih jezikih (Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Perl, PHP, Python, R, Ruby (vključno z Ruby on Rails), Scala, Clojure, Groovy, Scheme in Erlang). Odlikuje ga pregleden grafični vmesnik in možnost razširitev preko sistema vstavkov (ang. plugins) [16]. Sistem vstavkov nam omogoča dodajanje novih funkcionalnosti, ki jih osnovna aplikacija nima in služi prolagoditvi osnovne aplikacije lastnim potrebam.



Slika 3.4: Eclipse, razvojno okolje

Samsung Kies

Samsung Kies je brezplačna aplikacije, ki omogoča komunikacijo med novejšimi Samsungovimi pametnimi telefoni in operacijskim sistemom Windows ali Macintosh. Omogoča izdelovanje varnostnih kopij, prenos podatkov med telefonom in osebnim računalnikom, urejevanje večpredstavnostnih dokumentov in nadgradnje programske opreme pametnega telefona [17].

Android software development kit (SDK)

Razvojni programski paket Android SDK (Android Software development kit) vsebuje nabor orodij za razvoj aplikacij, ki tečejo na operacijskem sistemu Android OS [18].

3.5.2 Implementacija

Mobilna aplikacija obsega en Android projekt razdeljen na 6 aktivnosti (ang. activity), od katerih vsaka predstavlja svojo funkcionalnost skupaj s pogledom.

Podpora različnim vrstam zaslonov

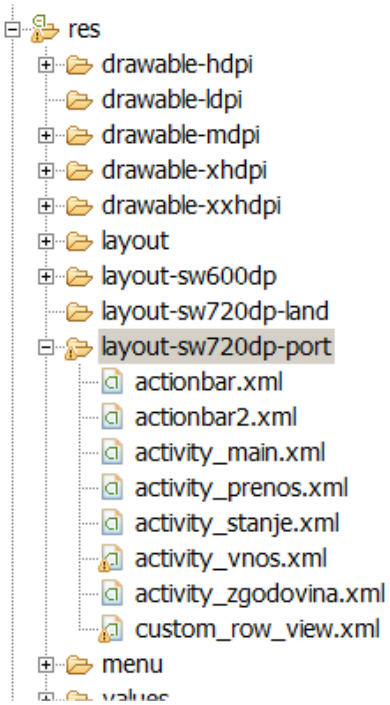
Grafična podoba aplikacije je sicer do neke mere avtomatsko prilagodljiva različnim ločljivostim in dimenzijam zaslonov, vendar je, glede na izdelane grafične elemente, najlepše prilagojena za dve konkretni vrsti zaslonov in sicer:

1. Zaslone ločljivosti 480x800 točk (Samsung Galaxy S3 Mini).

2. Zaslone ločljivosti 2560x1600 točk (tablica Samsung Nexus 10).

Za omenjena dva tipa zaslonov so, za vsako aktivnost, posebej definirane datoteke, ki definirajo postavitev elementov. Ravno tako so vsi slikovni elementi narejeni za obe vrsti zaslonov in je vsaka sličica v dveh variantah. Verzija za tablični računalnik ima povsod, kjer je možno, večjo pisavo, vse grafične

elemente večje ločljivosti in ponekod rahlo popravljeno razporeditev elementov. Android nam prilagajanje aplikacije različnim napravam omogoča na relativno enostaven način. Razpored elementov aplikacije se definira v navadni XML datoteki, ki je shranjena v direktoriju, imenovanem res in poddirektoriju imenovanem glede na značilnosti zaslona naprave (slika 3.5). Za zaslon, ločljivost in gostoto pik telefonov Samsung Galaxy S3 Mini, je razpored elementov definiran v privzetem direktoriju imenovanem layout, slike pa v direktoriju z imenom drawable-hdpi. Grafična podoba je bila narejena za



Slika 3.5: Definicija razporeditve za tablice

pokončni način uporabe aplikacije in ne za ležečega. Zato je v glavni nastavitveni datoteki aplikacije (AndroidManifest.xml) vsaka aktivnost omejena le na pokončen pogled.

Zahtevane funkcionalnosti in pravice naprave

Mobilna aplikacija od operacijskega sistema eksplicitno zahteva naslednje pravice za nemoteno delovanje:

1. android.permission.NFC - za dostop do NFC modula,
2. android.permission.INTERNET - za povezavo na internet,
3. android.permission.ACCESSWIFISTATE - za preverjanje stanja brezžične povezave,
4. android.permission.ACCESSNETWORKSTATE - za preverjanje stanja internetne povezave.

Od funkcionalnosti, ki jih mora mobilna naprava podpirati, je obvezen NFC modul. Sekcija s pravicami in zahtevanimi funkcionalnostmi se nahaja v konfiguracijski datoteki Android.xml.

Verzija operacijskega sistema

Minimalna verzija operacijskega sistema je nastavljena glede na uporabljene knjižnice iz ogrodja Android SDK (Android Software development kit). Najmanjša podprta verzija je 16, označena tudi kot Android 4.1 (Jelly Bean). To preverjanje izvede Eclipse ob prevajanju aplikacije in opozori na napako, če pogoj ni izpolnjen. Za primer se lahko v nastavitveni datoteki nastavi minimalno verzijo 14 in se ob prevajanju pojavi napaka (slika 3.6), ki pravi, da v tej verziji konstruktor NdefMessage z uporabljenimi parametri ne obstaja. Tako se moramo izogibati uporabi določenih klicev in razredov, oziroma potrebno funkcionalnost implementirati sami.

Preverjanje prisotnosti in pripravljenosti NFC

Preden se uporabi kakšen modul na mobilni napravi, je dobro preveriti ali mobilna naprava sploh ima fizični modul NFC in ali je vklopljen. Na začetku je potrebno s klicem funkcije PackageManager.hasSystemFeature poizvedeti

Slika 3.6: Minimalna verzija je Android 4.1

o prisotnosti določenega modula. Nato se modul inicializira in ugotovi v kakšnem stanju se nahaja.

Prenos zahtevka preko Android Beam

Osnovna ideja naloge je realizirati denarni prenos s pomočjo tehnologije NFC. To je na operacijskemu sistemu Android najlažje storiti z uporabo priložene funkcionalnosti Android Beam. Pogoj za uspešen prenos sta dve napravi, ki podpirata NFC in se nahajata v zbujenem stanju. Zato je potrebno najprej preveriti prisotnost in pripravljenost NFC modula. Ko je ta zagotovljena, se pripravi paket tipa NdefMessage, ki bo nosil željene podatke. V dogodku OnCreate() se nato pokliče metodo setNdefPushMessageCallback, ki lahko za razliko od metode setNdefPushMessage() ustvari NDEF sporočilo v odvisnosti od trenutnega konteksta aplikacije, kar je v našem primeru potrebno. Sprejemanje NDEF sporočila na drugi strani obsega branje vsebine na podlagi vnaprej znane vsebine binarnega zapisa in obdelavo sporočila.

Komunikacija s spletnimi servisi

Ker je komunikacija s spletnimi servisi lahko zamudno opravilo, je najbolje, da se izvaja v ozadju v ločeni niti. Temu je namenjen razred AsyncTask, ki je del knjižnic operacijskega sistema Android. AsyncTask je abstrakten razred, ki ustvari vzporedno nit, da ta opravi zadano nalogo in rezultat vrne nazaj na glavno nit. Izvajanje komunikacije s servisi v ločeni niti nam omogoča nemoteno delovanje uporabniškega vmesnika, medtem, ko aplikacija čaka na odgovor spletnih servisov. Razredu AsyncTask je potrebno definirati tip vhodnih in izhodnih podatkov in povoziti metodi doInBackground ter onPostExecute. Metoda doInBackground je tista, ki opravi klica na spletni servis in izvede prenos in pripravo podatkov za uporabo v aplikaciji. Metoda onPostExecute je namenjena vračanju rezultata na glavno nit. Začeti je potrebno s tem, da se definira razred BankSvcClient, ki implementira razred AsyncTask. V metodi doInBackground se izvede povezava na strežnik in branje ter razpoznava podatkov. Za povezavo na spletni servis se uporabi razred HttpClient in HttpResponse za branje rezultat zahtevka. Ko je zahtevek uspešno prebran, se pokliče metoda onPostExecute. Namen te metode je vračanje prebranih podatkov na glavno nit. To se naredi tako, da se preko reference na našo glavno aktivnost poišče kontrole in setira prebrane podatke.

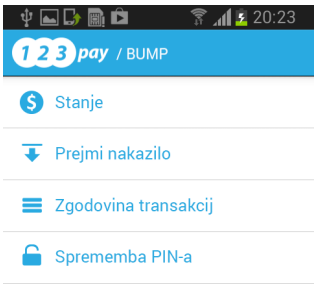
Uporabniški vmesnik

Uporabniški vmesnik je pregledno oblikovan in sestoji iz glavne strani in štirih podstrani. Glavna stran je hierarhično nadrejena podstranem, kar omogoča konsistentno navigacijo z gumbom gor, na kar so uporabniki operacijskega sistema Android navajeni.

Glavna stran - meni

Upravitelj glavne strani skrbi za meni na začetni strani (slika 3.7), ki se uporabniku prikaže, ko se aplikacija zažene. Naloga glavne strani je preveriti, ali je aplikacija zagnana prvič in uporabnik svoje PIN številke še ni vpisal.

Če se ugotovi, da v uporabniških nastavitvah PIN številke še ni, se uporabnika preusmeri na aktivnost strani za vnos. Brez PIN številke ne sme biti omogočena nobena funkcionalnost.



Slika 3.7: Glavni meni aplikacije

Vnos PIN številke

Upravitelj strani za vnos PIN številke skrbi za vnos in shranjevanje PIN številke v trajne nastavitve aplikacije (slika 3.8). Vsak uporabnik ima svojo PIN številko, ki je dodeljena samo njemu in ga tako kot identifikacijska številka računa, enolično določa. V našem prototipu se PIN številka shrani kar v nastavitve aplikacije, kjer ostane tudi pri naslednjem zagonu aplikacije. Funkcionalnost bi se lahko spremenila na tak način, da bi bilo potrebno PIN številko vnesti vsakič znova in s tem zagotovljeno dodatno varovanje pred morebitnimi nepridipravi.

Pregled stanja

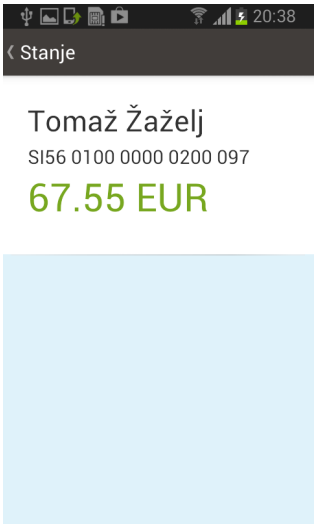
Upravitelj strani za pregled stanja skrbi za prikazovanje informacij o stanju na uporabnikovem računu in je dostopen le iz glavnega menija (slika 3.9).

Slika 3.8: Vnos številke PIN

Ob instanciranju upravitelja se v ločeni niti pošlje zahtevek z identifikacijsko številko računa na spletni servis. Spletni servis v odgovoru vrne informacijo o stanju na računu in IBAN številko transakcijskega računa. V našem prototipu aplikacije forma vsebuje osnovne podatke o imenu, IBAN številki računa in stanju računa. Za vrnitev nazaj na glavni meni je potrebno pritisniti gumb gor v glavi aplikacije. Stanje je v trenutni izvedbi aplikacije lahko le pozitivno, zato je barva zneska vedno zelene barve.

Pregled zgodovine prenosov

Ta stran prikazuje vse prenose, ki jih je uporabnik opravil (slika 3.10). V to spadata dve skupini prenosov. V prvi skupini uporabnik nastopa kot pošiljatelj, v drugi pa kot prejemnik denarja. Elementi prve skupine imajo znesek obarvan z rdečo barvo, kar nakazuje odliv z računa. Elementi druge skupine pa imajo znesek obarvan z zeleno barvo, kar nakazuje priliv na račun. Seznam prenosov je urejen po datumu, od zadnjega prenosa nazaj. Ker je seznam prenosov lahko dokaj dolg in bi lahko teoretično prišlo do večjega zamika pri nalaganju s spletnih servisov, je uporabljena grafika, ki simboli-



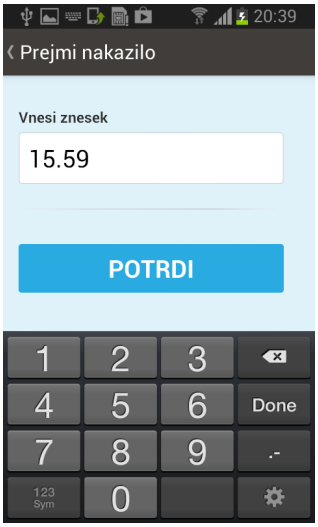
Slika 3.9: Pregled stanja

zira nalaganje podatkov. Le-ta se aktivira takoj ob instanciranju upravitelja strani in se odstrani, ko delovna nit v ozadju konča prenos podatkov in obvesti glavno nit.

Slika 3.10: Pregled zgodovine prenosov

Stran za začetek prenosa

Stran za začetek prenosa denarja omogoča vnos zneska, ki ga bo naprava preko storitve Android Beam sporočila drugi mobilni napravi (slika 3.11). Slednja bo nato preko spletnih servisov prenos dejansko izvedla. Forma vsebuje vnosno polje in gumb. Tipkovnica se avtomatsko aktivira v številčnem načinu. Ko uporabnik vnese znesek in pritisne gumb Potrdi, se aktivira upravitelj strani za prenos parametrov prenosa na drugo napravo.



Slika 3.11: Vnos zneska

Upravitelj prenosa

Upravitelj prenosa med dvema mobilnima napravama skrbi za prenos identifikacijske številke uporabniškega računa in zahtevanega zneska z ene naprave na drugo. Za prenos je uporabljena tehnologija NFC in storitev Android Beam. Preden se prenos lahko začne mora iniciator prenosa uporabiti stran za začetek prenosa, kjer vnese željeni znesek in pritisne potrditveni gumb. Naprava pošiljatelja nato uporabniku aplikacije izpiše navodilo, naj svojo mobilno napravo prisloni k napravi pošiljatelja zahtevanega zneska. Ko se napravi približata na razdaljo nekaj centimetrov in se vklopi NFC servis, pre-

vzame kontrolo operacijski sistem Android in od iniciatorja zahteva potrditev prenosa z dotikom zaslona. Ko se iniciator zaslona dotakne, se infromacija o identifikacijski številki uporabniškega računa in zahtevanem znesku prenese na telefon pošiljatelja denarja. Na telefonu pošiljatelja denarja operacijski sistem Android zazna prejem NFC paketa in odpre ustrezno aplikacijo, ki je morala prej biti registrirana za obdelavo tega paketa. Operacijski sistem na napravi pošiljatelja denarja aktivira direktno upravitelja prenosa in po prejemu parametrov s telefona prejemnika denarja prikaže potrditveno pojavno okno, ki uporabnika nagovori naj transakcijo potrdi ali zavrne. Če uporabnik transakcijo potrdi, se v ozadju kreira ločena nit in se na spletni servis pošlje zahtevek za prenos denarja z računa pošiljatelja na račun prejemnika v višini zahtevanega zneska. Če uporabnik transakcijo zavrne, se nič ne zgodi.

Prilagojena glava aplikacije

V nalogi je bilo potrebno prilagoditi glavo aplikacije (ang. Action bar) tako, da se jo pri vnosu PIN številke skrije, ker takrat ni dovoljena nobena druga akcija. Na strani glavnega menija leži posebna glava aplikacije, ki vsebuje logotip za označevanje aplikacije ali podjetja in nobenih drugih kontrol. Ker Android omogoče le delno prilagoditev glave aplikacije, je bilo potrebno v celoti nadomestiti privzeto glavo z novo. Potrebno je bilo narediti poseben razpored elementov v ločeni XML datoteki (za vsako vrsto zaslona ena datoteka). Celoten prostor glave pokriva slika v celostni grafični podobi podjetja skupaj z logotipom. Ob kreiranju pogleda je potrebno novo razporeditev elementov nastaviti na glavi. Na vseh ostalih straneh, razen na strani za vnos PIN številke in strani z glavnim menijem, je bila uporabljena nova glava, ki omogoča vrnitev na glavni meni in hkrati izpisuje naslov pogleda v točno določeni pisavi, sredinsko poravnano in na temni podlagi. Tudi to je najlažje izpeljati ravno s posebej prilagojeno razporeditvijo elementov. Ob kreiranju pogleda je potrebno razporeditev nastaviti, podati pravilni naslov in povezati gumb na glavi z metodo, ki uporabnika vrna na glavno stran.

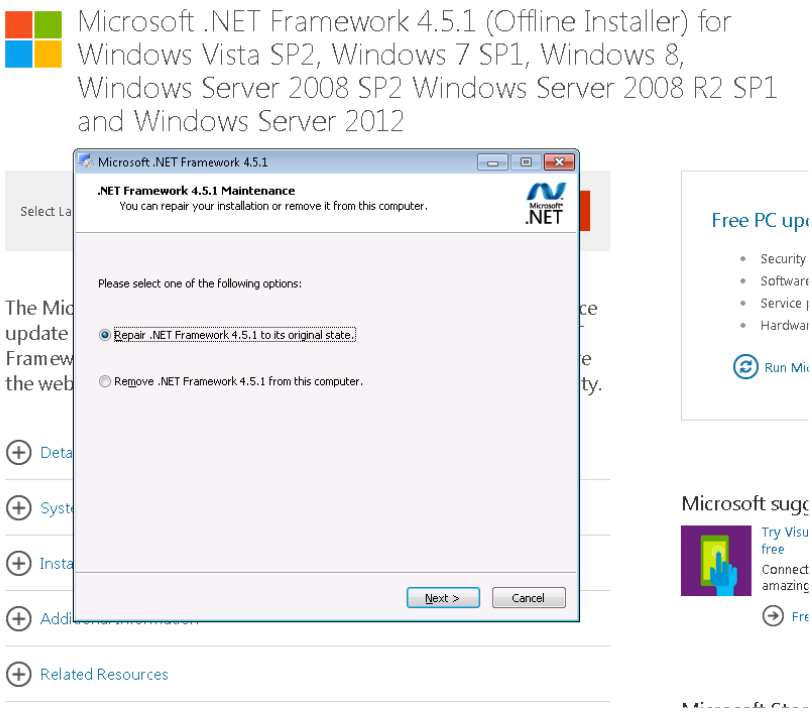
3.6 Testiranje aplikacije

3.6.1 Priprava podatkovne baze

Za podatkovno bazo je bila izbrana brezplačna različica Microsoftove baze SQL Express. Je prosto dostopna za prenos s spletnih straneh podjetja Microsoft. Za delo s podatkovno bazo je sicer na voljo aplikacija s tekstovnim uporabniškim vmesnikom, ki pride skupaj z bazo, vendar je tak pristop zamuden in neroden. Veliko lažje se je znajti v integriranem okolju SQL Managment Studio, ki omogoča vse administrativne in razvojne naloge nad podatkovno bazo. Po kreiranju baze, je bilo potrebno ustvariti tabele in v njih zapisati testne podatke. Na tabelah so kreirani tudi indeksi in nujne povezave med tabelo prenosov in računov. V pripravljene tabele je bilo potrebno vnesti še nekaj testnih uporabnikov za preverjanje delovanja aplikacije na čim bolj realnih podatkih. Ustvarjeno je bilo 5 različnih, izmišljenih uporabnikov. Vsak izmed njih ima določeno denarno stanje na računu in svojo PIN številko, s katero se lahko prijavi v aplikacijo in izvaja prenose in vpoglede v svoj račun. Na podatkovni bazi je bilo potrebno urediti dostop za aplikacijo in s tem je baza pripravljena za delo.

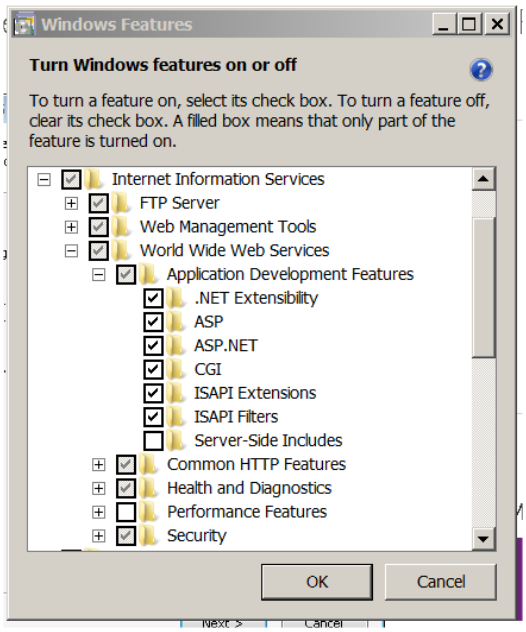
3.6.2 Priprava spletnih servisov

Ker so spletni servisi napisani v ogrodju .NET Framework 4.5, je bilo najprej potrebno namestiti zadnjo različico ogrodja na operacijski sistem (slika 3.12). Ogrodje je prosto dostopno s spletnih strani podjetja Microsoft, namestitev pa traja kar precej časa, posebej v primeru, da je na voljo le počasnejši računalnik. Ko je bilo ogrodje nameščeno, je bilo potrebno namestiti še vse potrebne komponente za poganjanje spletnih servisov v operacijskem sistemu Windows 7 Professional. Komponente so sicer priložene namestitvenemu paketu operacijskega sistema Microsoft Windows 7 Professional, vendar se ob namestitvi operacijskega sistema ne namestijo avtomatično. Obvezna komponenta je HTTP strežnik. Nameščen je bil Microsoftov strežnik, IIS

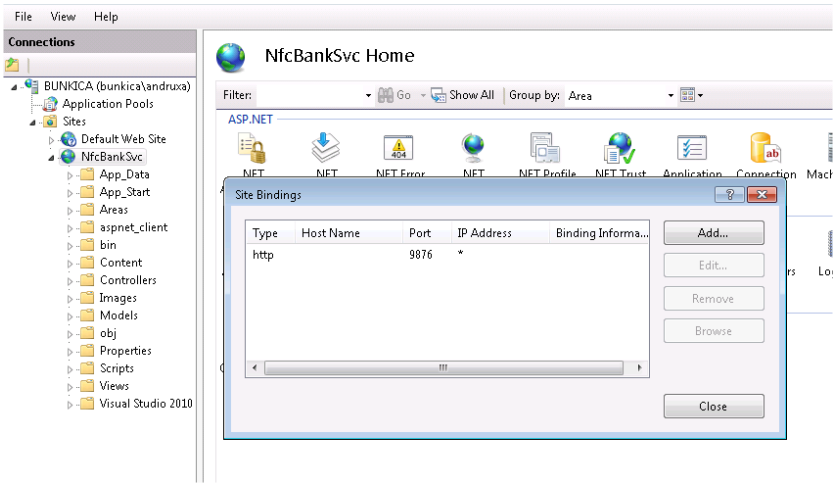


Slika 3.12: Namestitev ogrodja NET Framework

6.1(slika 3.13). Ko je strežnik uspešno nameščen, je potrebno v kontrolnem panelu ustvariti novo spletno aplikacijo, ji dodeliti ustrezne pravice in nastaviti vnaprej pripravljen direktorij aplikacije (slika 3.14). Ker gre za testno okolje, je bil strežnik postavljen na nestandardna vrata 9876, kar se je pri testiranju izkazalo za vir težav pri določenih operaterjih mobilne telefonije. V ta direktorij je nato potrebno prenesti vse potrebne datoteke aplikacije in nato še pravilno nastaviti nastavitve spletnih servisov. Ker je bila aplikacija zelo preprosta, je od nastavitev zahtevala le pravilno povezavo do podatkovne baze. Da je aplikacija pravilno nameščena in spletni servisi odzivni, je najlažje preveriti tako da se v brskalnik vpiše zahtevek za določeno metodo in če je vse vredu, se rezultat izpiše v XML obliki (slika 3.15).



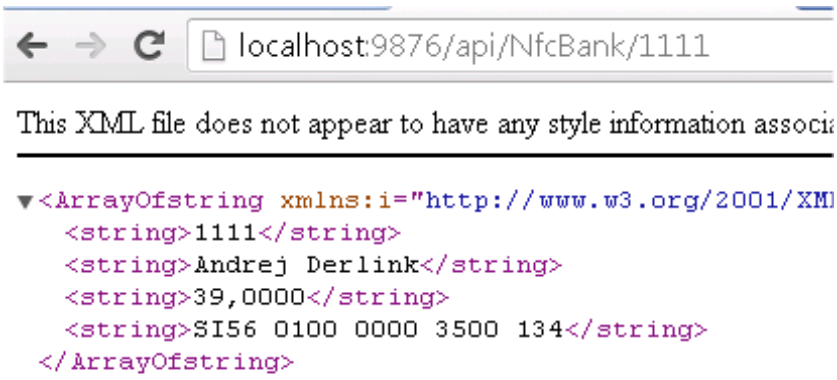
Slika 3.13: Namestitev HTTP strežnika IIS 6.1



Slika 3.14: Ustvarjanje aplikacije v strežniku IIS

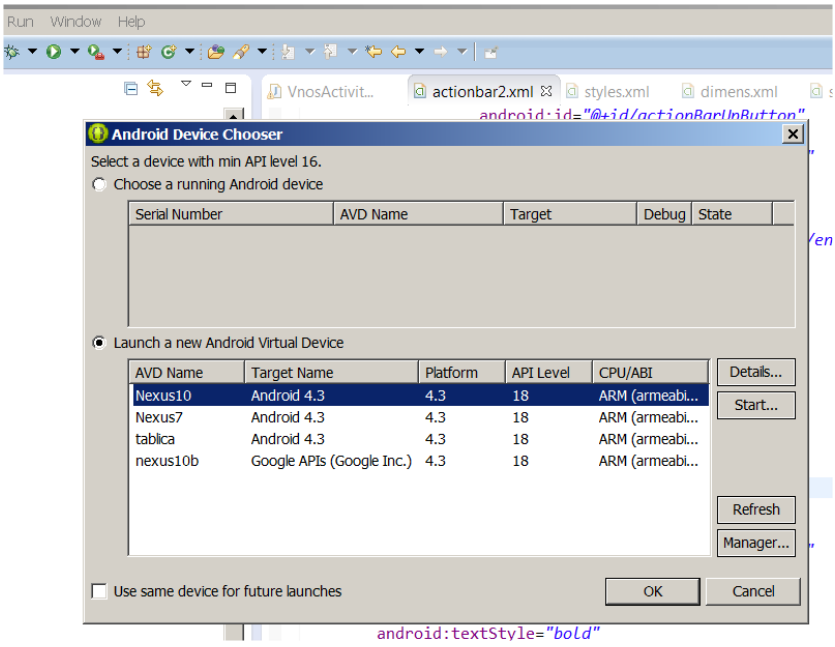
3.6.3 Priprava mobilnih naprav

Aplikacija je bila nameščena na dva mobilna telefona in tablični računalnik. Namestitev je potekala v razvojnem načinu, preko razvojnega okolja Eclipse.



Slika 3.15: Preverjanje ali se spletni servis odziva

Praktično celoten proces namestitve opravi Eclipse zgolj s pritiskom na gumb in izbiro ustrezne naprave (slika 3.16).



Slika 3.16: Namestitev mobilnih aplikacij

Poglavje 4

Sklepne ugotovitve

Mobilno aplikacijo za prenos denarnih nakazil smo testirali za različne uporabnike in preverili vse zahtevane funkcionalnosti (vpogled v stanje računa, pregled zgodovine prenosov in sam prenos denarja na transakcijskih računih med dvema uporabnikoma), kot tudi upravljanje prenosov s storitvijo Android Beam.

Prva zanimivost, ki je povezana z uporabo aplikacije, ne pa s predstavljeno rešitvijo, se je pokazala pri izbiri vrste omrežne povezave. Kadar je bila naprava povezana preko brezžičnega omrežja (WI-FI), so bili zahtevki na spletni servis normalno obdelani in aplikacija je delovala normalno. Ko pa je bila poskusno izključena brezžična povezava in se je naprava povezala preko mobilnega omrežja (3G), se je aplikacija pri enem od operaterjev mobilne telefonije začela obnašati nepredvidljivo. Rezultati spletnih servisov so bili napačne oblike, pojavljali so se nizi naključnih znakov. Ker se je to dogajalo le pri enem od operaterjev, je bilo najbolj smiselno posumiti, da je težava očitno v nestandardnih vratih, na katerih je poslušal HTTP strežnik. Začasno je bil HTTP strežnik premaknjen na standardna vrata za protokol HTTP, torej vrata 80 in vse je začelo delovati normalno. Tudi iskanje po spletu je obrodilo nekaj podobnih diskusij na forumih ljudi z enakimi težavami.

Naslednja težava pa je bolj povezana z zasnovo naloge in sicer gre za uporabniško izkušnjo prenosa podatkov z uporabo storitve Android Beam.

Ker Android Beam zahteva od uporabnikov potrditev prenosa z dotikom prsta, hkrati pa mora uporabnik napravo držati v neposredni bližini, oziroma v stiku z drugo napravo, se je to pokazalo za dokaj nehvaležno izvedbo. Potrebno je kar nekaj koncentracije, da uporabnik drži napravo staknjeno, medtem ko se dotakne s prstom zaslona, ne da bi vsaj za hip izmaknil napravo iz lege in prekinil prenos. Žal na to razvijalec nima vpliva, saj prenos preko Android Beam nadzoruje operacijski sistem in uporabniško aplikacijo določi šele sistem za razdeljevanje značk - kot je opisano v uvodnem delu naloge. Glede na relativno hiter razvoj tehnologije, je za pričakovati številne izboljšave v njenem razvoju, kakor tudi na področju uporabniških izkušenj. Po izvedbi testiranj in odpravljanju pomanjkljivosti je aplikacija delovala v skladu s pričakovanji in denarno nakazilo je bilo možno hitro in enostavno prenašati med računi uporabnikov navidezne spletne banke.

Ob uporabi razvite aplikacije in začetnimi težavami z mobilnim omrežjem 3G, iskanjem signala brezžičnega omrežja in nastavljanjem pravilnih naslovov HTTP strežnika se takoj pojavi vprašanje - kako izločiti uporabo strežnika. Ob vseh naštetih težavah je potrebno upoštevati tudi potovanja izven države, kjer naročnina ne krije prenosa podatkov in je uporabniku v odsotnosti brezžičnega omrežja dostop do interneta zelo drag. Prvi korak razširitve obstoječega sistema bi bila varna hramba podatkov na napravi sami. V ta namen je na nekaterih napravah že sedaj prisoten varni element, namenjen hranjenju občutljivih podatkov. V tem trenutku pa žal uporaba varnega elementa v uradni gradnji Androida ni podprta. Poti sicer obstajajo, vendar obsegajo spremembe v jedru operacijskega sistema, kar onemogoča izdelavo aplikacije za množično uporabo. Možna je seveda tudi uporaba navadnega pomnilnika v kombinaciji z enkripcijo. Takšna zaščita je bolj simbolična, saj se nezaščitene informacije vedno nahajajo v pomnilniku naprave in je vedno možna zloraba le-teh. Vendar pa lahko pričakujemo nadaljni razvoj na področju varnega shranjevanja podatkov na mobilnih napravah, kar bo omogočilo razvoj takšnih razširitev. V takem primeru bi potrebovali tudi način polnjenja računa in preverjanja avtentičnosti zapisanega stanja računa.

V nadaljevanju bi bilo mogoče razširiti aplikacijo na bolj ambiciozen način, kot naprimer uporabiti druge funkcionalnosti mobilnih naprav, kot so geolokacija in opozoriti uporabnika na vračilo dolga, ko bi se nahajal v neposredni bližini svojega upnika.

Literatura

[1] Vedat Coskun, Kerem Ok, Busra Ozdenizci *C Professional NFC Application Development for Android* Wrox, 2013.

[2] (2013) NFC Forum. Dostopno na:
`http://www.nfc-forum.org/home/`

[3] Massoud Moussavi *C Data Communication and Networking: A Practical Approach* Delmar, 2012.

[4] (2013) Nfc World. Dostopno na:
`http://www.nfcworld.com/nfc-phones-list/`

[5] (2013) Why Apple Doesn't Support NFC Payment on the iPhone Dostopno na:
`http://finance.yahoo.com/news/why-apple-doesnt-support-nfc-133400304.html`

[6] (2013) NDEF format. Dostopno na:
`http://members.nfc-forum.org/specs/spec_list/`

[7] (2013) NFC. Dostopno na:
`http://developer.android.com/guide/topics/connectivity/nfc/nfc.html`

[8] (2013) NFC Basics. Dostopno na:
`http://developer.android.com/guide/topics/connectivity/nfc/nfc.html`

[9] (2013) .NET Framework. Dostopno na:
http://www.microsoft.com/net

[10] (2013) ASP.NET MVC. Dostopno na:
http://www.asp.net/mvc

[11] (2013) ASP.NET Web API. Dostopno na:
http://www.asp.net/web-api

[12] (2013) LINQ to SQL. Dostopno na:
http://msdn.microsoft.com/en-us/library/bb386976(v=vs.110)
.aspx

[13] (2013) Microsoft Visual Studio. Dostopno na:
http://www.visualstudio.com/

[14] (2013) Microsoft SQL Express. Dostopno na:
http://www.microsoft.com/web/platform/database.aspx

[15] (2013) Java. Dostopno na:
https://www.java.com/en/

[16] (2013) Eclipse. Dostopno na:
http://www.eclipse.org/

[17] (2013) Samsung Kies. Dostopno na:
http://www.samsung.com/us/kies/

[18] (2013) Android SDK. Dostopno na:
http://developer.android.com/sdk/index.html

Slike