

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Gregor Mlakar

Implementacija interaktivne NFC aplikacije namenjene ogledu razstav

Diplomsko delo
na visokošolskem strokovnem študiju

Mentorica: doc. dr. Mira Trebar

Ljubljana, 2010



Št. naloge: 00477/2009

Datum: 15.10.2009

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

Kandidat: **GREGOR MLAKAR**

Naslov: **IMPLEMENTACIJA INTERAKTIVNE NFC APLIKACIJE NAMENJENE
OGLEDU RAZSTAVE**

**IMPLEMENTATION OF AN INTERACTIVE NFC APPLICATION
DESIGNED FOR EXHIBITION VISITS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Kandidat naj predstavi zasnovo in razvoj aplikacij, ki omogočajo uporabniku interaktivni ogled razstave. Diplomsko delo naj vključuje uporabo ustreznih brezžičnih tehnologij za povezavo računalnika z mobilno napravo in z označenim razstavnim eksponatom ter programskih orodij za razvoj in delovanje aplikacije. Implementacija naj bo izvedena z uporabo komunikacije v bližnjem polju (NFC - Near Field Communication) in preverjena z uporabo mobilnega telefona Nokia 6212 NFC na poenostavljenem modelu tehniškega muzeja v okviru laboratorija.

Mentor:


doc. dr. Mira Trebar



Dekan:


prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Gregor Mlakar,
z vpisno številko 63040267,

sem avtor diplomskega dela z naslovom:

Implementacija interaktivne NFC aplikacije namenjene ogledu razstav

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom doc. dr. Mire Trebar
- 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 _____ Podpis avtorja: _____

Zahvala

Zahvaljujem se mentorici, doc. dr. Miri Trebar za pomoč, svetovanje in usmerjanje pri izdelavi diplomske naloge. Zahvaljujem se tudi staršem za podporo, ki sem jo bil deležen v času izdelave diplomske naloge in tudi v času celotnega študija.

Kazalo

Povzetek	1
Abstract.....	3
1. Uvod	5
2. Uporabljene tehnologije	7
2.1 Identifikacija z radijskimi valovi (RFID)	7
2.2 Tehnologija NFC in njeno delovanje.....	12
2.3 Bluetooth.....	15
3. Uporabljena orodja in programska oprema pri razvoju aplikacije	21
3.1 Java	21
3.2 Eclipse in Eclipse ME.....	25
3.3 Sun Java 2 Micro Edition (J2ME) Wireless Toolkit za CLDC	26
3.4 XAMPP programski paket.....	27
3.5 Nokia PC Suite.....	28
3.6 Series 40 5. edition Nokia 6212 NFC SDK	29
3.7 Bluesoleil	30
4. Strojna oprema.....	31
4.1 Osebni računalnik z Bluetooth modulom	31
4.2 NFC mobilni telefon	32
4.3 RFID značke, za telefon NOKIA 6212 Classic NFC	34
5. Razvoj aplikacije za interaktiven ogled razstave.....	37
5.1 Podatkovna baza	39
5.2 Razvoj Bluetooth strežnika za PC.....	40
5.2.1 Bluetooth strežnik.....	40
5.2.2 MySQL Connector/J in uporaba MySql poizvedb	45
5.2.3 Uporaba niti	46
5.3 Razvoj aplikacije za mobilni telefon.....	47

5.3.1	Vmesnik za brezkontaktno komunikacijo (JSR 257).....	48
5.3.2	Bluetooth odjemalec	50
6.	Predstavitev aplikacij in namestitev.....	55
6.1	Aplikacije za interaktiven ogled razstav	55
6.2	Aplikacija za administracijo podatkovne baze in pregled statistike.....	56
6.3	Spletna stran	60
6.4	Namestitev aplikacij v realnem okolju.....	63
7.	Sklepne ugotovitve.....	65
7.1	Ocena dela	65
7.2	Ideje in funkcionalnosti za dopolnitev aplikacij.....	65
8.	Seznam uporabljenih virov	69

Seznam uporabljenih kratic in simbolov

Kratica in angleško ime	Opis
RFID Radio-frequency identification	Identifikacija z radijskimi valovi
NFC Near field communication	Komunikacija v bližnjem polju
SDK oz. JDK Software development kit Java Development kit	Zbirka že-napisanih knjižnic, ki jih lahko programer uporablja za razvoj lastnih programov oz. knjižnic
JVM Java Virtual Machine	Navidezni stroj Java
KVM Kilobyte Virtual Machine	Navidezni stroj Java, (omejitev 'Kilobyte')
JCP Java Community Process	Razvojna skupina, skozi katero gredo vse novosti, ki se pojavijo v Javi
JRE Java runtime environment	Java izvajalno okolje
J2ME Java 2 Platform Micro Edition	Java namenjena manj zmogljivim napravam
JSR Java specification request	Java specifikacija
JAD Java Application Descriptor	Datoteka, ki nosi opis Midlet aplikacije, ki je zapakirana v datoteki Jar.
JAR Java Archive	Arhiv v katerem so shranjene vse datoteke, ki so potrebne za izvajanje MIDlet programa in pripadajoči meta podatki
IDE Integrated development environment	Integrirano razvojno okolje
API Application programming interface	Aplikacijsko programirnljivi vmesnik
MIDP Mobile information device profile	Niz Java razredov, ki razširjajo funkcionalnost CLDC
CDC Connected device Configuration	Konfiguracija namenjena zahtevnejšim mobilnim napravam
CLDC Connected, limited device configuration	Konfiguracija namenjena najpreprostejšim mobilnim napravam
SQL Structured Query Language	Strukturirani povpraševalni jezik za delo s podatkovnimi bazami
WLAN Wireless local area network	Brezžično lokalno omrežje
WPAN Wireless personal area network	Brezžično osebno omrežje
GUI Graphical user interface	Grafični uporabniški vmesnik
Wi-Fi Wireless fidelity	Mednarodnem društvo Wi-Fi Alliance , ki podeljuje certifikate, da napravam narejenim po določenem standardu

ISO the International Organization for Standardization	Mednarodna organizacija za standardizacijo
ECMA International European Computer Manufacturers Association	Vodilni mednarodni organizacija na področju odprtih standardov
IEC the International Electrotechnical Commission	Mednarodna elektrotehniška komisija
EESC European Economic and Social Committee	EESO Evropski ekonomsko-socialni odbor
IEEE Institute of Electrical and Electronics Engineers	Inštitut inženirjev elektrotehnike in elektronike
EEPROM Electrically Erasable Programmable Read-Only Memory	Električno zbrisljiv in programirljiv bralni pomnilnik
IFF Identification friend or foe	Identifikacija zaveznik ali sovražnik
ISM band Industrial, scientific and medical radio bands	Frekvenčni pas namenjen Industrijski, znanstveni in medicinski uporabi
OMA Open Mobile Alliance	Organizacije, katere cilj je povečati trg mobilnih storitev
GNU GPL GNU General Public License	Licenca za prosto dostopno programsko opremo
PDA Personal digital assistant	Dlančnik oziroma osebni digitalni pomočnik
OTA Over the air	Prenos s spleta po zraku
EDR Enhanced Data Rate	Razširjen prenos podatkov
AVCTP Audio / video control transport protocol	Protokol za nadzor avdio in video signalov
AVDTP Audio / video distribution transport protocol	Protokol za distribucijo avdio in video signalov
IC Integrated circuit	Integrirano vezje
SC Smart cards	Pametne značke, kartice, oznake
UID Unique identification number	Unikatna serijska številka
UUID Universal Unique identification number	Univerzalna unikatna serijska številka
AES Advanced encryption standard	Trenutno najbolj razširjen algoritem za šifriranje podatkov
DES Data encryption standard	Algoritem za šifriranje podatkov
NXP Next eXPeriance Semiconductors	Ime za družbo ki izdeluje polprevodnike osnovana s strani Philipsa
GPS Global Positioning System	Sistem globalnega določanja položaja
AFH Adaptive Frequency Hopping	Mehanizem preskakovanja frekvenc

XHTML Extensible Hyper Text Markup Language	Označevalni jezik, ki ima enak namen kot HTML, vendar je usklajen s sintakso XML
CSS Cascading Style Sheets	Predloge, ki določajo vizualno podobo spletnih strani
XML Extensible Markup Language	Preprost razširljiv označevalni jezik, ki nam omogoča format za opisovanje strukturiranih podatkov
PHP Hypertext Preprocessor oziroma Personal Home Page Tools	Orodja za osebno spletno stran

Povzetek

Z uporabo ustreznih orodij in programske opreme je implementirana aplikacija, ki s kombinacijo tehnologij NFC (ang. Near Field Communication) in Bluetooth služi za interaktiven ogled razstav v muzejih oziroma galerijah. Poleg tega je implementirana aplikacija, ki je namenjena vnosu razstavnih eksponatov in oddelkov v podatkovno bazo ter pregledu statistike. Zgrajena pa je bila tudi spletna stran, ki je namenjena predstavitvi muzeja.

Delo obsega predstavitev tehnologij, orodij ter programske in strojne opreme, ki so bila uporabljena pri razvoju aplikacij. Predstavljena je tehnologija RFID, NFC in Bluetooth, orodja Eclipse, XAMPP, Nokia SDK, Java in nekatera druga. Opisana je uporabljena strojna oprema: računalnik z Bluetooth modulom, mobilni telefon Nokia 6212 Classic in RFID značke. Sledi predstavitev podatkovne baze ter zasnova in potek razvoja aplikacij. V zaključku so predstavljene razvite aplikacije, podane pa so tudi ideje za njihovo dopolnitev in ocena dela.

Osnovni cilji naloge so:

- Implementirati aplikacijo, za interaktiven ogled razstav, ki bo vključevala:
 - Aplikacijo za mobilni telefon, ki bi lahko izboljšala uporabniško izkušnjo ob obisku muzeja ali galerije. Aplikacija bo z uporabo NFC tehnologije, prebrala podatke z RFID značke in se preko ene izmed obstoječih brezžičnih tehnologij (WLAN, Bluetooth, mobilna omrežja) povezala na strežnik, ki bo tekel na osebnem računalniku. Strežniku bo posredovala unikatno identifikacijsko številko RFID značke, nato pa sprejela podatke o razstavnem eksponatu in jih prikazala.
 - Aplikacijo za osebni računalnik, ki bo delovala kot strežnik in sprejemala zahteve odjemalcev in jim posredovala zahtevane podatke. Strežnik bo zasnovan tako, da sprejme unikatno identifikacijsko številko, izvede poizvedbo v podatkovni bazi in odjemalcu posreduje podatke (tekst, sliko), ki se nanašajo na prejeto identifikacijsko številko. Strežnik skrbi tudi za zapis vsake poizvedbe v posebno tabelo v podatkovni bazi, ki se uporabi za izračun statistike.
- Implementirati aplikacijo, ki bo služila kot grafični vmesnik za vpis razstavnih eksponatov in oddelkov, pregled in nastavljanje statusa oddelkov in za izračun ter pregled statistike.
- Razvoj spletne strani, ki bo služila osnovni predstavitvi muzeja, za prikaz podatkov pa bo uporabljala isto podatkovno bazo, kot pri ostalih aplikacijah.

Ključne besede: komunikacija NFC, Bluetooth, značke RFID, Java, MIDlet, XAMPP.

Abstract

This thesis presents an application to link together Near Field Communication (NFC) and Bluetooth technology and will be used for an interactive tour of exhibitions in museums and galleries. Besides that an application for entering exhibits and departments information into database and for review of the statistics is implemented. Additionally, a web site for the presentation of the museum was also build.

The diploma work comprehend the study and presentation of technologies, tools, software and hardware equipment, that were used during applications development. The presentations of RFID, NFC and Bluetooth technologies, Eclipse, XAMPP, Nokia SDK, Java and others followed with the description of hardware: PC with a Bluetooth module, mobile phone Nokia 6212 Classic and RFID tags are included. The database system and the description of applications development with a simplified museum scenario and new ideas for their upgrade and work assessment are presented.

Basic goals of the thesis:

- Implementation of an application for interactive tour of exhibitions, which involves:
 - Client application for the mobile phone that could improve the user's experience while visiting museums or galleries. Application will use NFC technology to read data from RFID tags and connect it to the server by one of the existing wireless technologies (WLAN, Bluetooth, mobile networks) that will run on a PC. Client will send UID number to the server and wait for the response (data, image) and display it.
 - Desktop application defined as a server which will accept client's request and respond with the appropriate information. The server will receive the unique identification number, make an SQL Query in the database and send the data (description, image) from the database to the client. The task of the server will also be to record any query into appropriate table in the database that will be used to calculate the statistics.
- Implementation of GUI application for entering exhibits and department's information into database, review, adjust the status of departments and to calculate and review statistics.
- Development of a website, which will be used for presentation of the museum on the web. Data shown by browser will be a part of the same database which is used for other applications.

Keywords: NFC communication, Bluetooth, RFID tags, Java, MIDlet, XAMPP.

1. Uvod

Razvoj tehnologij vse bolj vpliva na naše vsakdanje življenje, saj se z njimi srečujemo na vsakem koraku. Na področju povezovanja, komuniciranja in izmenjave informacij se danes vse pogosteje uporabljajo različne brezžične tehnologije, ki jih v vse večjem številu pričakujemo kot osnovni del elektronskih naprav. Veliko število mobilnih naprav podpira različne brezžične tehnologije, kot so: Wi-Fi, Bluetooth, GPS, GSM, UMTS, EGPRS. Tehnologije se ločijo po doseženem dometu, porabi energije, hitrosti prenosa informacij in po namenu uporabe.

Najpogosteje uporabljena brezžična omrežja so javno dostopna mobilna omrežja, ki sodijo v skupino WWAN (ang. Wireless Wide Area Network) omrežij. Uporablja se jih za prenos zvoka in podatkov na večjih razdaljah (do 3 km od bazne postaje). V to skupino sodijo tehnologije GSM (ang. Global System for Mobile communication), UMTS (ang. Universal Mobile Telecommunication System), EGPRS (ang. Enhanced General Packet Radio Service) in HSDPA (ang. High Speed Downlink Packet Access). Sledijo jim brezžična lokalna omrežja (ang. Wireless Local Area Network), ki v omejenem dosegu (od 100 m znotraj, do 300 m zunaj) omogočajo povezovanje dveh ali več elektronskih naprav. Uporabljajo se za dostop do obstoječih Ethernet omrežij in interneta.

Prvi široko uporabljen standard oziroma oblika brezžičnega lokalnega omrežja WLAN (ang. Wireless Local Area Network) je bil standard Wi-Fi (802.11). Standard določa format podatkov, prenosne frekvence in komunikacijske protokole. Sledili so mu standardi 802.11 a , 802.11 b, 802.11 g, 802.11 n. Trenutno najpopularnejša sta standarda 802.11 b in 802.11 g, ki za komunikacijo uporabljata 2.4 GHz ISM frekvenčni pas. 802.11 b ima maksimalen prenos podatkov omejen na 11 Mbps, 802.11 g pa ima maksimalen prenos podatkov omejen na 54 Mbps in je kompatibilen z napravami 802.11 b.

Naslednja brezžična tehnologija, ki se je v zadnjem času široko uveljavila je Bluetooth. Prvotno je bila načrtovana za brezžično povezovanje elektronskih naprav na majhnih razdaljah in za zamenjavo žičnih povezav med posameznimi perifernimi enotami. Njen namen pa se je kmalu razširil tudi na področje brezžičnih lokalnih omrežij. Tehnologija Bluetooth se uvršča med osebna brezžična omrežja kratkega dometa WPAN (ang. Wireless Personal Area Network). Lastnosti so prenos podatkov do 3Mbps, nizka poraba energije, domet pa je odvisen od energijskega razreda v katerega naprava spada (od 4 do 100 m).

Pred nekaj leti pa se je na trgu pojavila nova brezžična tehnologija NFC (ang. Near Field Communication), ki si počasi utira pot med množico že uveljavljenih tehnologij. Pojavlja se predvsem na področju mobilnih telefonov in omogoča enostavno, brezkontaktno komunikacijo kratkega dometa, med elektronsko napravo in RFID značko. V diplomski nalogi so predstavljene lastnosti omenjene tehnologije, prikaz možnih načinov uporabe in realizacija enega izmed njih.

Cilj diplomskega dela je z uporabo obstoječih orodij in tehnologij razviti programsko opremo, ki bi jo v muzejih ali galerijah uporabljali za interaktiven ogled razstav in oddelkov, ki se v njih nahajajo. Kot osnova pri razvoju programske opreme bo uporabljen poenostavljen primer tehniškega muzeja, ki je razdeljen v dve nadstropji in pet oddelkov.

Razviti je bilo potrebno:

- Aplikacijo za PC, ki bo delovala kot strežnik. Namen aplikacije je, da odjemalcem preko Bluetooth povezave posreduje ustrezne podatke iz podatkovne baze tehničnega muzeja, hkrati pa vse poizvedbe zapisuje v ustrezno tabelo, ki se nahaja v isti podatkovni bazi.
- Aplikacijo za mobilni telefon, ki z RFID značke prebere unikatno identifikacijsko številko, jo preko Bluetooth povezave pošlje strežniku in počaka odgovor. Ko strežnik odgovori z opisom in sliko razstavnega eksponata, aplikacija na mobilnem telefonu poskrbi za prikaz podatkov na zaslonu.
- Razvoj aplikacije za PC, ki služi kot grafični vmesnik, za vpis oddelkov in razstavnih eksponatov, pregled in nastavljanje statusa posameznim oddelkom ter za izračun in pregled statistike. V okviru statistike se vodi podatke o obiskovalcih in njihovem številu, ogledanih razstavnih eksponatih in njihovi priljubljenosti.
- Razvoj spletne strani, ki je namenjena osnovni predstavitvi muzeja in prikazu podatkov o oddelkih in razstavnih eksponatih, ki se nahajajo v njih. Podatki se prikažejo iz podatkovne baze tehničnega muzeja.

Za dosego ciljev je bilo potrebno:

- Pregledati ponudbo mobilnih naprav, ki podpirajo tehnologijo NFC in vsaj enega izmed komunikacijskih protokolov (Wi-Fi, Bluetooth).
- Izbrati ustrezno brezžično tehnologijo za prenos podatkov med strežnikom in mobilno napravo.
- Preučiti komunikaciji NFC in Bluetooth.
- Izbrati ustrezno podatkovno strukturo za hranjenje informacij.
- Preučiti komunikacijo s podatkovno strukturo.
- Izbrati ustrezno platformo za razvoj NFC in Bluetooth aplikacij.
- Izbrati ustrezna orodja za razvoj NFC in Bluetooth aplikacij.
- Implementirati izbrane komunikacijske protokole v aplikaciji za mobilni telefon.
- Implementirati izbrane komunikacijske protokole v aplikaciji za PC (ang. Personal computer).
- Implementirati aplikacijo za administracijo podatkovne baze in pregled statistike.
- Zgraditi spletno stran muzeja, ki dinamično prikaže podatke o razstavnih eksponatih iz podatkovne baze.

2. Uporabljene tehnologije

2.1 Identifikacija z radijskimi valovi (RFID)

Zgodovina

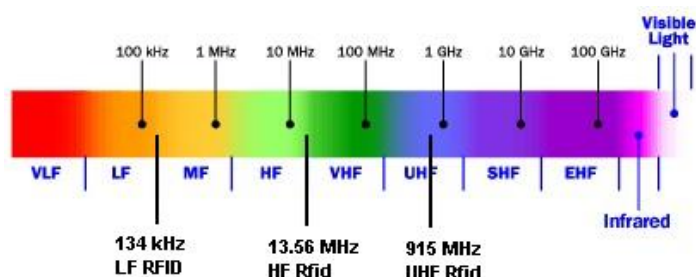
Začetek razvoja tehnologije sega v čas 2. svetovne vojne, ko so v vojni sodelujoče velesile uporabljale škotsko tehnologijo iz leta 1935 imenovano radar. Na podlagi radarja ni bilo mogoče identificirati komu pripada zaznano letalo. V ta namen so Britanci razvili prvo aktivno identifikacijo IFF (ang. Identification friend or foe), ki je zavezniško letalo ločilo od sovražnikovega. To so storili tako, da so Britanska letala opremili z oddajnikom, ki ga je radar preko povratnega signala spoznal za zavezniško. V kolikor letalo ni poslalo pravega signala, so letalo označili za sovražnikovo.

Tehnologija RFID (ang. Radio-frequency identification), kot jo poznamo danes, se je razvijala med letoma 1960 in 1980. Prve komercialne aktivnosti so se pojavile v začetku šestdesetih let. Takrat so razvili elektronski nadzor opreme za preprečevanje krajev. Okoli leta 1970, so v ZDA tehnologijo uporabljali za avtomatizacijo tovarn, sledenje vozil in dostop do varovanih območij. V Evropi pa se je v preteklih desetletjih, tehnologija večinoma uporabljala v aplikacijah, kjer je nujna brezkontaktna identifikacija (kontrola dostopa, avtomatično cestninjenje, sledenje živalim, itd.)

V devetdesetih letih, se je z željo po hitrejšem, zanesljivejšem, bolj avtomatiziranem in poenotenem sistemu označevanja ter sledenja različnih produktov, tehnologija RFID vse bolj uveljavljala. Največja ovira je bila cena, ki pa s časom postaja vse bolj dostopna. Tako se bodo RFID sistemi, najverjetneje na različnih področjih uporabljali vse pogosteje.

RFID Frekvence

Na področju identifikacije z radijskimi valovi so uporabljena tri frekvenčna področja, ki so določena s standardom v posameznih državah in unijah. Slika 1 prikazuje radio frekvenčni spekter od 100 kHz do 100 GHz in na njem označene frekvenčne pasove.



Slika 1. Frekvence RFID.

(Vir: Internet: <http://static.howstuffworks.com/gif/radio-spectrum.gif>)

Nizke frekvence LF (ang. Low Frequency): od 9 do 135 kHz

- Najmanj občutljive za branje v primeru prisotnosti kovin in tekočin,
- Uporabljajo se večinoma s pasivnimi značkami,
- Počasna hitrost branja,
- Domet od 0 cm (fizičen kontakt) do 0.5 metra,
- Večinoma se uporabljajo za kontrolo dostopa, identifikacijo živali ter za sledenje posod z odpadki in plinskih jeklenk.

Visoke frekvence HF (ang. High Frequency): 13.56 MHz

- Rahlo občutljiva na prisotnost kovin in prevodnih tekočin,
- Večinoma pasivne značke,
- Srednja hitrost branja,
- Domet do enega metra,
- Uporabljajo se za označevanje knjig v knjižnicah in biometrične podatke.

Ultra visoke frekvence UHF (ang. Ultra High Frequency): 868 do 915 MHz

- Občutljive so na prisotnost kovin, dobro pa prodirajo skozi neprevodne materiale in neprevodne tekočine,
- Aktivne in pasivne značke,
- Hitro branje,
- Domet tudi do 100 metrov,
- Uporabljajo se večinoma v preskrbovalnih verigah.

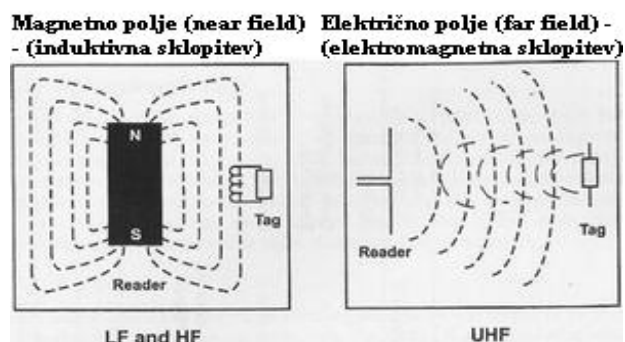
RFID sistem

Identifikacija z radijskimi valovi je brezžična komunikacijska tehnologija, ki uporabnikom zagotavlja unikatno identifikacijo označenih objektov [1]. Slika 2 prikazuje sistem RFID, ki je sestavljen iz čitalca RFID in značke. Značka nosi informacijo, s katero unikatno določa predmet, na katerega je pritrjena. Čitalec pa z oddajanjem elektromagnetnega valovanja ali oddajanjem magnetnega polja to informacijo prebere.



Slika 2. Čitalec in značka RFID.

Glede na način napajanja oddajnika, se RFID sistemi delijo na pasivni, polpasivni in aktivni RFID. Glede na način sklopitve prek radijskih valov, pa na induktivno in elektromagnetno sklopljene sisteme (slika 3).



Slika 3. Načina sklopitev RFID čitalnika in značke.

- Induktivni RFID sistem za prenos informacije uporablja princip magnetne indukcije. Dve bližnji tuljavi sta induktivno sklopljeni in ko magnetni pretok, ki ga povzroča tok prve tuljave, doseže drugo tuljavo na njenih priključnih sponkah inducira napetost. Komunikacija deluje v bližnjem polju (ang. Near Field), zato je domet induktivnih sistemov do nekaj deset centimetrov in pada z naraščanjem frekvence (krajšanje valovne dolžine). Oddajnik informacijo prenese čitalcu, prek uporabe bremenske modulacije (ang. Load Modulation), ki v praksi pomeni spreminjanje sklopnega faktorja med tuljavama v ritmu podatkov. Induktivni RFID sistemi za delovanje najpogosteje uporabljajo nizke (LF) in visoke frekvence (HF).
- Elektromagnetni RFID sistem komunicira z uporabo elektromagnetnih (EM) valov. Čitalec oddaja EM valove, ki dosežejo oddajnik in se od njega odbijejo. Ta odboj lahko izkoristimo za prenos informacije od oddajnika do čitalca. V trenutku, ko se oddajnik vzbudi, začne svojo lastno impedanco spreminjati v ritmu podatkov in na ta način spreminja svojo oscilacijsko frekvenco. Signal, ki se odbije je moduliran, celoten pojav pa imenujemo modulacijski odboj (angl. Modulation Backscatter). Elektromagnetno sklopljeni RFID sistemi se običajno uporabljajo za daljše razdalje. Frekvence, ki jih ti sistemi uporabljajo so ultra-visoke UHF in tudi super visoke frekvence SHF (ang. Super High Frequency).

RFID značke

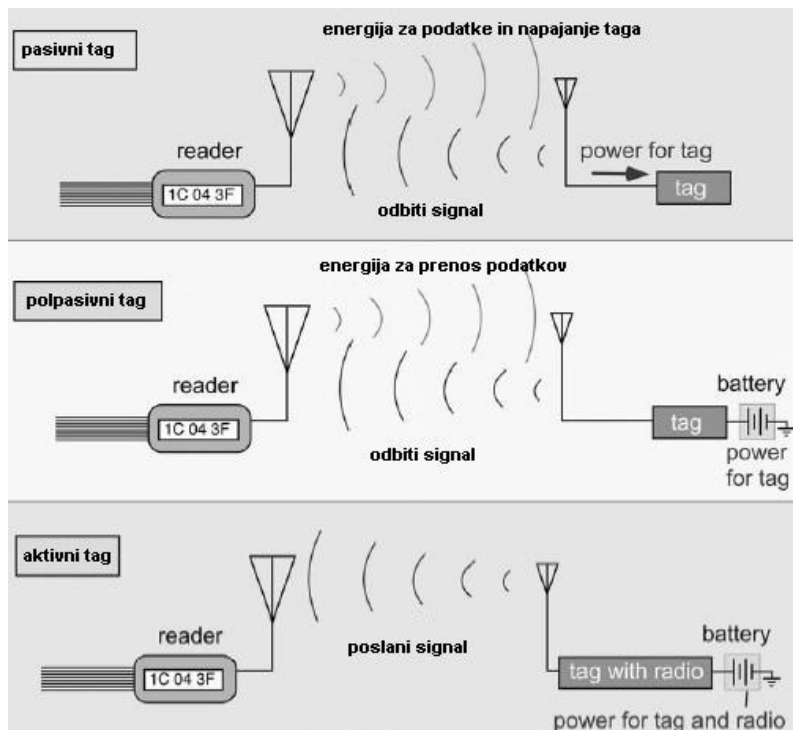
RFID značka ali RFID oddajnik, je sestavljen iz integriranega vezja in antene. Integrirano vezje hrani podatke in izvaja modulacijo in demodulacijo signalov. Antena pa je namenjena sprejemanju in oddajanju radijskih signalov.

Z vidika napajanja (slika 4), se značke delijo na:

- Aktivne značke poleg antene in čipa vsebujejo tudi baterijo, ki napaja vezje. Aktivne značke so večje in dražje od polpasivnih in pasivnih, imajo pa številne prednosti. Te so večja moč oddajanja, daljši domet (do 100 metrov) ter bolj zanesljivo delovanje v neprijaznem okolju (bližina kovine ali tekočine). Imenujemo jih tudi pametne značke,

saj lahko nanje priključimo senzorje za merjenje hitrosti, določanje lege, merjenje temperature in vlage. Življenjska doba aktivnih značk je nekaj let.

- Polpasivne značke ravno tako vsebujejo baterijo. Od aktivnih značk se razlikujejo po tem, da za oddajanje signala uporabljajo napajanje radijskega signala. Zanesljivost branja je enaka kot pri aktivnih značkah, doseg pa je enak kot pri pasivnih. Življenjska doba je do 10 let. Branje aktivnih in polpasivnih značk poteka pri frekvencah do 868 MHz.
- Pasivne značke ne potrebujejo lastnega napajanja, temveč potrebno energijo za delovanje, dobijo neposredno od signala, ki se inducira v anteni. Iz antene sprejet izmenični signal se usmeri in dovede do čipa, ki se vzbudi in prične z delovanjem. Pasivne značke imajo nižjo ceno (od 7 do 20 centov) ter manjše dimenzije od aktivnih, vendar posledično veliko krajši doseg (do 5 metrov) in manjšo odpornost na napake. Življenjska doba je nekaj desetletij.



Slika 4. Možne oblike RFID značk, glede na način napajanja.
(Vir.S.Shepard, RFID (McGraw-Hill, New York, 2005))

Poleg omenjenih razlik, se značke razlikujejo še glede na frekvenco delovanja, velikost pomnilnika ter glede na namen uporabe. Najpreprostejše značke vsebujejo samo UID (ang. Unique Identifier), kar pomeni 64 oziroma 96 bitni blok podatkov, ki je narejen samo za branje. Naprednejše pa vsebujejo pomnilnik za branje in pisanje ter ustrezne kriptografske algoritme za varovanje informacij shranjenih v njem.

Glede na zunanji videz, se RFID značke pojavljajo v najrazličnejših oblikah. Lahko so kartice, etikete, nalepke, obeski, ploščice ali škatlice. Odvisno od namena za katerega se bodo uporabljale.

Prednosti RFID značk:

- V primerjavi s črtnimi kodami velja, da je ta tehnologija imuna na umazane ter prebarvane značke, saj se informacije prenašajo preko radijskih valov. Radijski valovi, pa z lahkoto potujejo čez materiale, kot so papir, barva, umazanija.
- Za razliko od črtnih kod, je kopiranje RFID značke praktično nemogoče, zato je uporaba značk priporočljiva, kadar so podatki o osebah ali predmetih zaupnejše narave.
- Črtne kode morajo biti vidne in fizično dostopne čitalcem, medtem ko čitalec lahko RFID značko prebere tudi v notranjosti embalaže.
- RFID značke se da prilagoditi zahtevnemu industrijskemu okolju, kot so ekstremna temperatura, vlaga, umazanija, jedke tekočine, mehanska odpornost.
- Možno je hkratno branje velikega števila značk v dosegu RFID čitalca.
- Imajo dolgo življenjsko dobo, saj ne vsebujejo nobenih premičnih delov.
- Prenos podatkov se izvrši v nekaj milisekundah.

RFID čitalci

Čitalci v okolje oddajajo radijske valove in poslušajo odgovore, ki jih pošiljajo RFID značke. To nam omogoča identifikacijo predmetov na daljavo. Ko RFID značka vstopi v območje dosega čitalca, ta preko ustrezne antene, s pomočjo radijskih valov prebere oziroma po potrebi spreminja podatke, ki so zapisani na znački. Prvi RFID čitalci so bili zasnovani za branje ene vrste značk, danes pa so vse pogostejše v uporabi čitalci, ki omogočajo branje več vrst značk. Protokolizirski protokol jim omogoča, da lahko hkrati berejo večje število značk v svojem dosegu. Obstajajo v različnih velikostih in izvedbah. Med večje spada čitalec z več antenami, priklopljen na osebni računalnik. Srednje veliki so podobni dlančnikom, najmanjši pa so v velikosti poštne znamke in so primerni za vgradnjo v mobilne telefone.

Primeri uporabe:

- V skladišču z vpeljano RFID tehnologijo, lažje vodijo dejansko količino zaloge. RFID oddajne nalepke so nalepljene na paleto (lahko tudi na posamezni artikel znotraj palete), čitalci tovrstnih nalepk, pa so pravilno razporejeni po prostoru tako, da zaznajo nalepke v vsakem položaju.
- V zabaviščnih centrih ali igralnicah, bi se z uporabo RFID igralnih žetonov, ki imajo vgrajeno vezje za oddajanje identifikacijske številke, lahko preprečilo unovčenje ponarejenih igralnih žetonov. V igralnici bi lahko s pomočjo čitalnikov prepoznali žetone, ki ne oddajajo pravega signala.
- V knjižnicah z RFID tehnologijo je možno na lažji način voditi evidenco knjig in ostalega gradiva. Omogoča popolnoma avtomatizirano delovanje celotne knjižnice, saj si lahko stranke knjižnice gradivo izposojajo in vračajo same. Zmogljivi sortirni sistemi lahko na podlagi informacije, ki jo vsebuje nalepka, sortirajo knjige po nadstropjih ali oddelkih ter na ta način zmanjšajo obremenitev zaposlenih v knjižnici.
- Z uporabo tehnologije RFID, je možno izvesti inventuro po policah veliko hitreje kot s črtno kodo, saj artiklov ni potrebno jemati s polic, temveč se le s čitalcem sprehodimo med policami.
- Registracija delovnega časa: v številnih podjetjih registrirajo prihode in odhode zaposlencev na osnovi RFID sistema.
- RFID značka, ki se jo vsadi v telo živali, lahko vsebuje podatke o živali, lastniku in njenem zdravstvenem stanju.

- Kontrola dostopa, ki omogoča prehod v poslovno stavbo ali hišo. Čitalnik je nameščen pred vrati, ki jih lahko odklenemo, le z ustrezno RFID kartico oziroma značko.
- Elektronsko cestninjenje.
- V trgovini bi lahko nakupovalno košarico postavili na ustrezno mesto. RFID čitalnik pa bi samodejno ugotovil seznam predmetov v košarici in izpisal račun.

Povezava med RFID in NFC

Identifikacija z radijskimi valovi in komunikacija v bližnjem polju NFC (ang. Near Field Communication) sta sorodni brezžični tehnologiji. RFID je tehnologija v prvi vrsti namenjena sledenju, identifikaciji in upravljanju s produkti, živalimi ali ljudmi. NFC pa je tehnologija izpeljana iz RFID sistema, ki ima doomet komunikacije omejen na do 10 cm in je v prvi vrsti namenjena manjšim denarnim transakcijam (npr. plačevanju hrane, pijače in javnega prevoza) in aplikacijam, kjer je prisotna večja potreba po varnosti.

Razlog za pojavitev NFC tehnologije je bila predvsem potreba po varovanju zaupnih podatkov. Radijsko-frekvenčni valovi, če imajo na voljo dovolj energije, nudijo prenos informacij na zelo dolgih razdaljah. Prenos informacij na dolge razdalje pa želimo imeti samo pri nekaterih aplikacijah (npr. sledenje živalim), medtem ko pri aplikacijah, kjer se uporabljajo osebni podatki ali denarne transakcije, dolgega dometa ne želimo, saj daljši kot je doomet, bolj verjetna je zloraba zaupnih informacij.

2.2 Tehnologija NFC in njeno delovanje

Začetek razvoja tehnologije sega v leto 2002, ko sta podjetji Sony Corporation in Royal Philips Electronics želeli spodbuditi razvoj NFC po vsem svetu. Z medsebojnim sodelovanjem, sta pripravili osnutek specifikacij, ki sta ga razvili na podlagi svojih lastniških protokolov Felica in Mifare. Predložili sta ga organizaciji odgovorni za standardizacijo informacijskih in komunikacijskih sistemov ECMA International. ECMA (ang. European Computer Manufacturers Association) je osnutek leta 2004 potrdila kot standard ECMA-340 in ga naknadno predložila v odobritev organizaciji ISO/IEC (ang. International Organization for Standardization/International Electrotechnical Commission), ki je standard potrdila pod oznako ISO/IEC IS 18092. Za nadaljnji razvoj tehnologije skrbijo v okviru NFC Foruma, katerega člani so proizvajalci, prodajalci, ponudniki storitev in ostala podjetja, ki se na kakršenkoli način srečujejo s to tehnologijo [2].

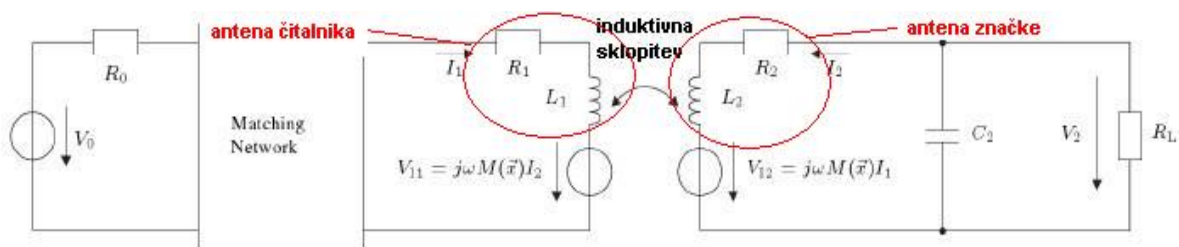
Komunikacija v bližnjem polju (NFC), je brezžična, povezavna tehnologija, ki omogoča enostavno brezkontaktno komunikacijo kratkega dometa med elektronsko napravo in RFID značko, lahko pa tudi med različnimi elektronskimi napravami. Tehnologija NFC je enostavna za uporabo, hitra in varna. Te lastnosti ji omogočajo uporabo v številnih aplikacijah, kot so elektronsko plačevanje in druge denarne transakcije.

Osnove delovanja

NFC komunikacija deluje v bližnjem magnetnem polju in za delovanje uporablja frekvenco 13.56 MHz. Komunikacija med napravami oziroma med značko in napravo se vzpostavi, ko se nahajata na razdalji do 10 cm, sama hitrost prenosa pa znaša do 424 Kbit/s. NFC naprave omogočajo združljivost z ostalimi brezžičnimi tehnologijami, kot so Wi-Fi, Bluetooth in

GSM omrežja (GPRS, UMTS, HSDPA), zato lahko NFC deluje kot začetna ali kot končna točka v povezovalnih aplikacijah [2].

Tehnološka osnova komunikacije v bližnjem polju, je radiofrekvenčna komunikacija na podlagi magnetne indukcije. Anteni značke in čitalca NFC, ki se nahajata dovolj blizu skupaj, tvorita magnetni sklop. Preprost model induktivno sklopljenih tuljav je prikazan na sliki 5. Levo vezje predstavlja NFC čitalec, desno pa značko. Anteni sta predstavljeni z oznako za tuljavo. L_1 in L_2 sta lastni induktivnosti obeh tuljav, ki sta povezani z medsebojno induktivnostjo M . Čitalec skozi tuljavo pošilja izmenični tok, zaradi česar v bližnji okolici nastane izmenično magnetno polje. Ko v to polje vstavimo drugo manjšo tuljavo, se v njej inducira izmenična napetost. Napetost se uporabi za napajanje in s tem tudi posredovanje podatkov iz značke na čitalec, saj se tudi na tuljavi značke inducira napetost, ki čitalcu odgovori z bremenom integriranega vezja. To breme pa predstavlja podatke, ki se nahajajo na znački RFID [3].



Slika 5. Model induktivne sklopitve med čitalnikom in oznako.
(Vir. Journal of Communications, vol. 2, no. 6, November 2007)

Načini komunikacije

Podobno kot pri RFID sistemih, tudi pri NFC obstajata dva načina komunikacije:

- **Pasivni način:** način komunikacije, pri katerem aktivna naprava oddaja nosilni signal. Pasivna naprava ga modulira in na ta način preko induktivnega sklopa, posreduje svoj odgovor aktivni napravi. Pri tem ni nujno, da se pasivna naprava napaja s signalom aktivne, torej je lahko pasivna naprava tudi polpasivna. Pomembno je le, da komunikacijo začne aktivna naprava.
- **Aktivni način:** pri tem načinu obe napravi izmenično oddajata svoja signala, kar pomeni, da se obe napravi napajata preko svojega lastnega vira energije. To je aktiven RFID sistem, v katerem sta obe komunicirajoči napravi aktivni v smislu napajanja in oddajanja signala.

Varnost:

- Kratek doomet komunikacije, čeprav to ni zanesljiva zaščita, saj bi napadalec z ustrežno opremo komunikaciji prisluškoval na razdalji tudi do več metrov.
- Za 'varen' prenos podatkov med dvema NFC napravama, je potrebno vpeljati kodiran kanal, s pomočjo simetričnih šifrirnih postopkov z varno izmenjavo ključa.
- Uporaba značk, ki za varovanje podatkov uporabljajo napredne šifrirne algoritme.

Uporabnost

Tehnologija NFC se danes pojavlja na področju mobilnih telefonov (slika 6). Napovedi kažejo, da naj bi bila v roku 3 do 5 let tretjina mobilnih telefonov opremljenih z NFC čipom [4]. Kasneje, pa naj bi se NFC čipi pojavljali tudi v drugih elektronskih napravah, kot so digitalni fotoaparati, kamere, televizije, dlančniki, osebni računalniki in celo v gospodinjskih aparatih.



Slika 6. NFC telefon Benq T80 in RFID značka.

V svetu dandanes poteka veliko pilotskih projektov, ki poskušajo dokazati uporabnost tehnologije, tako za uporabnike, kot za podjetja. Možnosti za uporabo NFC tehnologije so praktično neomejene, katere ideje pa se bodo razvile v uporabne aplikacije, pa bo pokazal čas. Aplikacije lahko razdelimo na štiri osnovne tipe, ki se razlikujejo po načinu, na katerega uporabnik sproži dogodek [2] :

- 'Touch and go': uporabnik NFC napravo približa drugi NFC napravi ali RFID znački, nakar se samodejno izvede neka operacija. Primer aplikacij: nadzor dostopa, elektronska vstopnica, elektronska vozovnica in prenos podatkov s pametnih plakatov (ang. Smart poster).
- 'Touch and confirm': uporabnik mora operacijo pred izvedbo potrditi, bodisi prek vnosa ustreznega gesla ali s pritiskom na tipko "OK". Primer aplikacije: mobilno plačevanje.
- 'Touch and connect': povezava dveh naprav NFC, kar lahko izkoristimo za medsebojni prenos raznih podatkov, kot so slike, avdio datoteke, vizitke.
- 'Touch and explore': NFC naprava nudi več kot eno možno funkcijo. Uporabnik ima na voljo več možnosti, med katerimi izbira.

Največji potencial za uporabo NFC tehnologije v prihodnosti, je na področju:

- Vozovnic in elektronskih vstopnic: uporabnik bi si lahko na NFC telefon prenesel elektronsko vstopnico (za javni prevoz, parkirno hišo, prireditve) in jo uporabil kot plačilno sredstvo, pri uporabljanju določene storitve. Račun pa bi poravnal prek svojega mobilnega operaterja. (Primer: London in Frankfurt izvajata testna projekta, za uporabo NFC telefonov, v javnem transportnem omrežju)
- Oglaševanje: NFC naprave so lahko sredstvo za nov način oglaševanja. Osnovna ideja je tako imenovani "pametni plakat (ang. Smart poster)", ki je opremljen z RFID značko,

ki vsebuje osnovne informacije o plakatu, in si jih lahko uporabnik z NFC telefonom enostavno prenese in shrani na telefon.

- Povezljivost z ostalimi napravami: prednosti NFC sta predvsem hitrost vzpostavitve komunikacije, ki je veliko hitrejša kot pri Bluetooth in enostavnost uporabe (ne potrebuje ročnega vzpostavljanja povezave). Zato je primerna za hiter prenos manjših datotek, kot so vizitke, slike ali imeniki med dvema mobilnima napravama.

2.3 Bluetooth

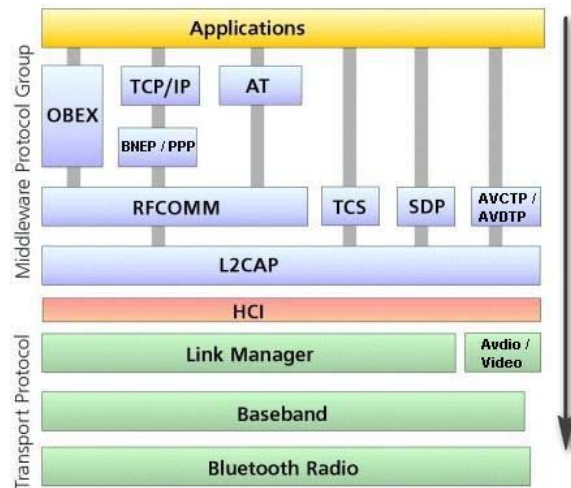
Je brezžična komunikacijska tehnologija, ki so jo razvili v Ericssonovih laboratorijih leta 1994. Njihov cilj je bil razviti nizko cenovno brezžično tehnologijo, namenjeno prenosu podatkov in govora pri elektronskih napravah, kot so telefoni, prenosni računalniki in dlančniki. Bila je ena izmed najbolj pričakovanih standardov, saj je obljubljala brezžično povezovanje poljubnih elektronskih naprav.

Prvi rod Bluetooth naprav, je imel veliko težav z delovanjem. Težave pri povezovanju so imele celo naprave istega proizvajalca, če pa smo želeli povezati napravi različnih proizvajalcev, komunikacija skoraj zagotovo ni delovala. S časom se je tehnologija izpopolnila ter postala ena izmed bolj razširjenih. Druga generacija Bluetooth, je prinesla hitreše prenose ter zanesljivejšo in hitrejšo povezljivost. V ta namen so se pojavili Bluetooth profili, ki poskušajo zagotoviti združljivost ter usklajeno delovanje naprav različnih proizvajalcev.

Bluetooth komunikacija ima arhitekturo odjemalec/strežnik. Strežnik čaka odjemalca, da poišče napravo in servis, nato pa mu odgovori z ustreznimi podatki. Bluetooth dosega hitrosti do 3 Mb/s, pri tem pa porabi vsega 1/100 moči Wi-Fi. Deluje v pasovni širini ISM (ang. Industrial, scientific and medical radio bands) od 2400 MHz do 2483,5 MHz. V tem spektru se nahajajo še drugi protokoli (npr. Wi-Fi), zato Bluetooth z namenom, da se naprave med seboj ne motijo, uporablja mehanizem preskakovanja frekvenc AFH (ang. Adaptive Frequency Hopping). Mehanizem zaznava že zasedene kanale in se jim po potrebi izogne. Frekvenčno območje razdeli na 78 kanalov in po njih preskakuje s hitrostjo 1600 skokov na sekundo. Za pravilno medsebojno komunikacijo morajo naprave upoštevati pravila komuniciranja, ustrezati Bluetooth standardu, predvsem pa morajo biti narejene v skladu z Bluetooth specifikacijo, ki je sestavljena iz Bluetooth sklada in Bluetooth profilov.

Bluetooth sklad

Je strojna programska oprema (ang. Firmware). Upravlja z nastavitvami, komunikacijskimi parametri in močjo signala Bluetooth naprave. Sestavljen je iz plasti (slika 7), ki imajo pri delovanju točno določeno vlogo. Proizvajalci Bluetooth naprav, običajno ne uporabljajo vseh plasti, saj se naprave razlikujejo po samem namenu delovanja. Zgornje plasti do HCI, so integrirane v aplikaciji ali tečejo kot gonilniki. Spadajo pod aplikacijski nivo. Spodnje plasti se izvajajo v Bluetooth modulu, ki skrbi za komunikacijo med Bluetooth napravami in spadajo pod transportni nivo [5].



Slika 7. Bluetooth sklad in plasti.

(Vir: Internet: http://cgi.cse.unsw.edu.au/~cs4411/wiki/images/a/a9/Bluetooth_stack.jpg)

Transportni nivo:

Link Manager - skrbi za varnost in nadzira povezave ter porabo energije (nadzira moč oddajanja signala).

Baseband - ima eno izmed najpomembnejših funkcij. Skrbi za sinhronizacijo, časovno naslavljanje in preskakovanje frekvenc. Upravlja podatkovne pakete in jih po potrebi popravlja. Dve povezani napravi, morata biti usklajeni tako, da frekvenco menjavata istočasno ter medtem sprejemata in pošiljata podatkovne pakete.

Bluetooth Radio - definira zahteve, ki jih potrebuje Bluetooth oddajnik v povezovalnem pasu 2,4 GHz ISM ter upravlja funkcije radia (ojača, modulira, demodulira signal, sintetizira in spreminja frekvence, pretvarja bite v simbole).

Aplikacijski nivo:

HCI (ang. Host controller interface) – je vmesnik med našo in gostiteljevo Bluetooth napravo.

L2CAP (ang. Logical link controller adaptation protocol) – plast, ki skrbi za deljenje vseh podatkov (podatke, ki jih dobi razdeli pravim profilom), segmentiranje in sestavljanje paketov, ki potujejo skozi enoto. Sprejema 64KB dolge pakete iz višjih nivojev in jih razdeli v okvirje primerne za prenos ali obratno. Podatke pa nato posreduje nižji ali višji plasti.

RFCOMM (ang. Virtual serial port protocol) - je plast, ki simulira serijski prenos podatkov (RS 232) po radijskih frekvencah ter je osnova za storitve, ki so vezane na zaporedni vmesnik npr. prenos datotek

SDP (ang. Service discovery protocol) - plast, ki se uporablja za iskanje storitev na oddaljenih Bluetooth napravah. Sama komunikacija v bistvu poteka med SDP strežnikom in SDP odjemalcem.

OBEX (ang. Object exchange protocol) - je protokol za izmenjavo podatkov, kot so datoteke in je implementiran nad RFCOMM plastjo. OBEX push protokol, je namenjen potiskanju datotek na telefon. OBEX ftp pa je namenjen ftp povezavi preko Bluetooth medija.

AVCTP (ang. Audio / video control transport protocol) in AVDTP (ang. Audio / video distribution transport protocol) - protokola za nadzor in distribucijo avdio in video signalov.

BNEP (ang. Bluetooth network encapsulation protocol) - je protokol, ki Ethernet promet ovija v Bluetooth okvirje. Z uporabo te plasti, se lahko uporablja druge povezavne protokole, kot so IP, TCP, UDP.

Bluetooth profili

Bluetooth profili so programski vmesniki, ki neposredno skrbijo za komunikacijo s protokoli na nižjih nivojih, posredno pa za zanesljivejšo komunikacijo med napravami. V Bluetooth skladu, jih uvrščamo pod aplikacijske protokole. S profili, ki jih ima določena naprava implementirane, je določeno katere storitve ponuja. Bluetooth opredeljuje široko paleto profilov, ki opisujejo več različnih primerov uporabe. Z upoštevanjem pravil v Bluetooth specifikaciji, lahko razvijalci ustvarijo aplikacije, skladne z aplikacijami, ki tečejo na drugih napravah. Nekaj najpogosteje uporabljenih profilov je opisanih v tabeli 1.

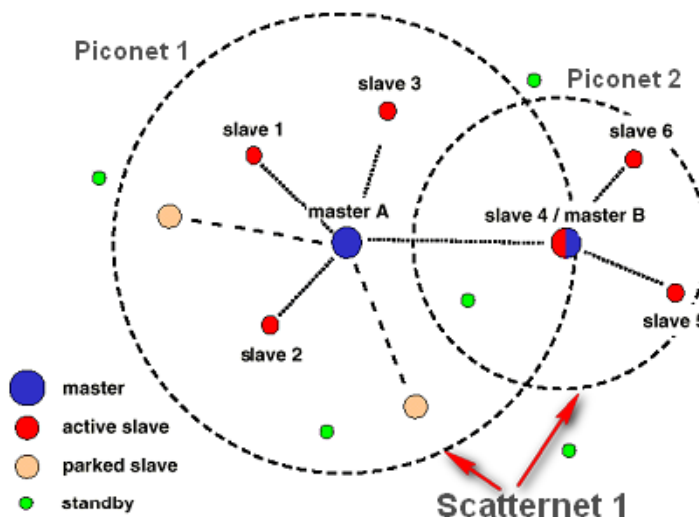
Ime profila:	Opis
Advanced Audio Distribution Profile (A2DP)	Za prenos stereo zvoka
Audio / Video Control Transport Protocol (AVRCP)	Za oddaljeni nadzor TV ali Hi fi sistema
Basic Imaging Profile (BIP)	Za prenos slik, ter oddaljeni nadzor tiskalnika
Basic Printing Profile (BPP)	Za tiskanje teksta, slik, elektronske pošte
Common ISDN Access Profile (CIP)	Za prenos ISDN signala
Cordless Telephony Profile (CTP)	Za uporabo naprave kot brezžični telefon
Dial-Up Network Profile (DUN)	Za klicni dostop do interneta
File Transfer Profile (FTP)	Za dostop do datotečnega sistema druge naprave
General Audio/Video Distribution Profile (GAVDP)	Osnova za A2DP in VDP profila
Generic Object Profile (GOEP)	Za prenos datoteke med napravama
Hands-Free Profile (HFP)	Za prostoročno telefoniranje
Hard Copy Cable Replacement Profile (HCRP)	Za tiskanje
Headset Profile (HSP)	Za povezavo naprave in slušalk
Human Interface Device Profile (HID)	Za povezavo s tipkovnico, miško
Intercom Profile (ICP)	Za glasovno povezovanje naprav, brez uporabe telefonskih omrežij
Object Push Profile (OPP)	Za pošiljanje objektov med napravami
Personal Area Networking Profile (PAN)	Za vzpostavljanje omrežij
Service Discovery Application Profile (SDAP)	Za odkrivanje servisov na oddaljeni napravi
Service Port Profile (SPP)	Za serijsko povezavo med napravami
Synchronization Profile (SYNC)	Za sinhronizacijo imenika, podatkov
Video Distribution Profile (VDP)	Za prenos video signala

Tabela 1. Bluetooth profili.

Bluetooth profili so lahko implementirani v kateremkoli programskem jeziku. So neodvisni od platforme, saj se Bluetooth profil nanaša na določeno vrsto funkcionalnosti za napravo Bluetooth. Tako je lahko profil implementiran na Palm OS v C++ ali v zbirnem jeziku na Bluetooth tiskalniku. Če želimo, da komunikacija preko Bluetooth povezave pri uporabi določene storitve deluje brez težav, morata obe napravi podpirati in imeti ustrezno prilagojen Bluetooth profil ter imeti implementiran Bluetooth sklad. Na primer, če želimo dve napravi medsebojno sinhronizirati, morata obe podpirati sinhronizacijski profil ali če želiš iz ene naprave na drugo poslati datoteko, morata obe napravi podpirati Object push profil.

Bluetooth omrežje

Bluetooth tehnologija je bila načrtovana za standardizirano povezovanje elektronskih naprav na majhnih razdaljah in za zamenjavo žičnih povezav med posameznimi perifernimi enotami. Njen namen pa se je kmalu začel širiti tudi na področje brezžičnih omrežij. Uvrščamo jo med osebna brezžična omrežja kratkega dometa WPAN (ang. Wireless Personal Area Network). Omogoča izgradnjo majhnih zasebnih omrežij, ki se imenujejo piconet. Piconet omrežje je sestavljeno iz naprave, ki ima vlogo gospodarja ter do 7 naprav, ki imajo vlogo sužnja. Gospodar je naprava, ki nastavi uro, preskakovanje frekvenc in dostopno kodo. Piconet omrežja se povezujejo v scatternet omrežja (slika 8), ki jih lahko sestavlja do 10 piconet omrežij ali do 80 naprav, nato pa omrežje postane nasičeno [6].



Slika 8. Prikaz Bluetooth omrežja.

(Vir: Internet: <http://www.forum.nokia.com/piazza/wiki/images/1/19/Piconet.png>)

Poleg omenjenega povezovanja naprav, pa lahko s pomočjo Bluetooth pristopnih točk, podobno kot pri Wi-Fi, vzpostavimo Bluetooth omrežje, ki ima dostop do lokalne mreže ter s tem do podatkov, ki se nahajajo na lokalnih strežnikih.

Varnost

Standard Bluetooth zagotavlja več stopenj varnosti:

- Izmenjava ključev med napravama med povezovanjem,
- Vsako Bluetooth napravo je mogoče narediti za nevidno,
- Že vzpostavljena zveza je zaščitena z (največ) 128-bitnim kodiranjem,
- Hitro menjavanje frekvenc (1600-krat na sekundo zamenja frekvenco),
- Doseg naprav je odvisen on energijskega razreda (tabela 2):

Energijski razred	Poraba moči	Doseg
Razred 1	100 mW	100 m
Razred 2	2.5 mW	10m
Razred 3	1mW	1m

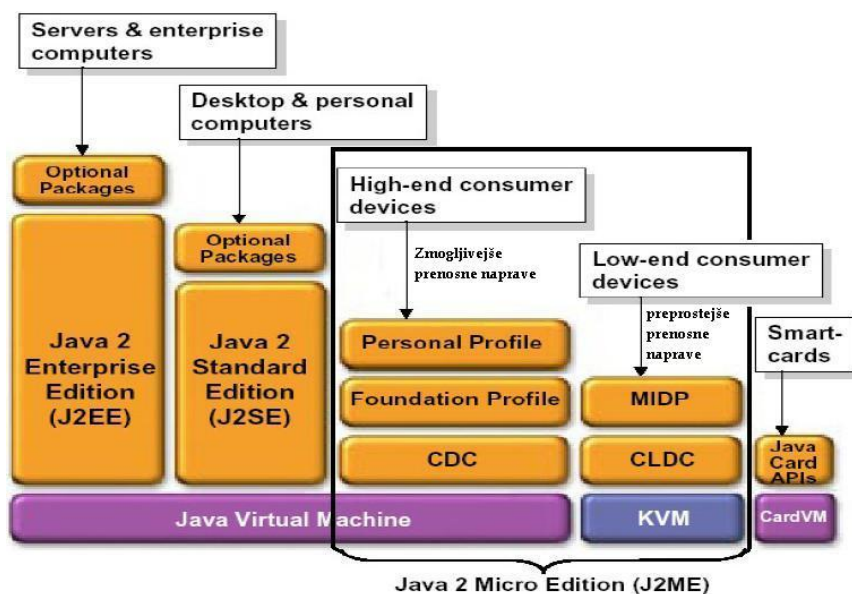
Tabela 2. Energijski razredi Bluetooth naprav.

3. Uporabljena orodja in programska oprema pri razvoju aplikacije

To poglavje je namenjeno predstavitvi orodij in programske opreme, ki so bila uporabljena pri razvoju in testiranju aplikacij. Vsa ta orodja so odprtokodna in prosto dostopna na spletu.

3.1 Java

Java je visokonivojski, objektno orientiran programski jezik 3. generacije. Razvili so jo pri podjetju Sun Microsystems in leta 1996 predstavili njeno prvo različico, ki ni bila deležna pravega uspeha. Bistveni preskok, je leta 1998 prinesla različica 1.2, ki je bila kasneje preimenovana v Java 2 SE (ang. Standard Edition) različica 1.2. Poleg standardne različice, je Java 1.2 prinesla še nekoliko okrnjeno Java ME (ang. Micro Edition) in tudi izpopolnjeno različico Java EE (ang. Enterprise Edition). Java ME je namenjena manj zmogljivim napravam (mobilni telefoni, dlančniki), medtem ko je Java EE namenjena programskim strežnikom. Prikaz delitev Java 2 platforme je prikazan na sliki 9.



Slika 9. Grafični prikaz delitve Java 2 platforme.

(Vir: Internet: <http://codewurd.files.wordpress.com/2009/09/1231.jpg>)

Do danes, se je Java 2 razvila v enega najpopularnejših in najpogosteje uporabljenih programskih jezikov. Njene pozitivne lastnosti so enostavnost, objektna usmerjenost, neodvisnost od platforme, trdoživost, široka razširjenost, varnost, dinamičnost in visoka zmogljivost. Java kot programski jezik ni nič posebnega. Razlog za uspešnost, je predvsem v njenem izvajanjem okolju ter v programskih knjižnicah [7].

Za razvoj Jave in programskih knjižnic od različice 1.4 naprej, skrbijo v okviru razvojne skupine JCP (Java community process), skozi katerega gredo vse novosti, ki se v Javi pojavijo. Skupina JCP skrbi za standardizacijo specifikacij in tehnologij na platformi Java. Sestavljajo jo predstavniki podjetja Sun in drugi kupci javanske tehnologije (podjetje IBM,

HP, Oracle, Borland, Bea, Nokia, Philips, Siemens, Sony, Cisco, Ericsson,...). Namen podjetij, ki so vključena v to skupino, je ohraniti čim večjo prenosljivost kode med napravami različnih proizvajalcev. Njihova glavna naloga je odločanje o tem, kateri JSR (ang. Java Specification Requests) bodo sprejeti kot novi programski vmesniki API.

Izvajalno okolje JRE (ang. Java Runtime Environment), ki poskrbi, da je Java neodvisna od platforme, je bila prava mala revolucija. Uveljavljeni programski jeziki, so programsko kodo prevajali neposredno v strojno kodo, kar sicer omogoča hitrejše izvajanje, vendar v primeru, da ni popolna, predstavlja nevarnost za sistem. Programska koda v Javi, pa se izvaja v nadzorovanem okolju in se ne prevaja v strojno, ampak v vmesno kodo (bytecode). Program preveden v vmesno kodo, se v primeru, da zanj obstaja ustrezen tolmač (interpreter), lahko izvaja na poljubni napravi. Tolmaču v Javi rečemo tudi javanski navidezni stroj JVM (ang. Java Virtual Machine) in je mnogo več, kot samo navaden tolmač. Poleg osnovne funkcije interpretiranja programske kode, Java programom ponuja nadzorovano izvajalno okolje, ki skrbi za nekatere temeljne storitve (npr. samodejno sproščanje pomnilnika, varnost pred prekoračitvijo medpomnilnika) [7].

Za osnovni razvoj Java programov potrebujemo razvojni paket JDK (ang. Java Development kit), katerega vsebina je predstavljena v tabeli 3 in preprost urejevalnik besedila. Namesto urejevalnika besedila, lahko posežemo po razvojnih orodjih IDE (ang. Integrated Development Environment), ki nam delo močno olajšajo (npr. Eclipse, NetBeans, JBuilder). Sveže različice razvojnih paketov za operacijske sisteme Windows, Solaris in Linux in postopki njihovih namestitvev, so na voljo na spletnih straneh Jave.

Podmapa:	Vsebina
bin	orodja JDK(prevajalnik, ...)
demo	demonstracijski programčki in programčki v Javi
include	datoteke, potrebne za povezavo Jave z izvorno strojno kodo
jre	izvajalno okolje
lib	knjižnice

Tabela 3. Podmape, ki jih vsebuje paket JDK.

Izdelava Java programa, po namestitvi paketa JDK, poteka v treh korakih:

1. Korak: Kodo napišemo v poljubnem urejevalniku besedil ali v razvojnem orodju in jo nato shranimo v datoteko s končnico .java (npr. Hello_World.java).
2. Korak: Napisano kodo prevedemo s prevajalnikom javac (javac Hello_World.java). S tem dobimo vmesno kodo s končnico .class (npr. Hello_World.class).
3. Korak: Prevedeno vmesno kodo interpretiramo z JVM (java Hello_World).

Java ME

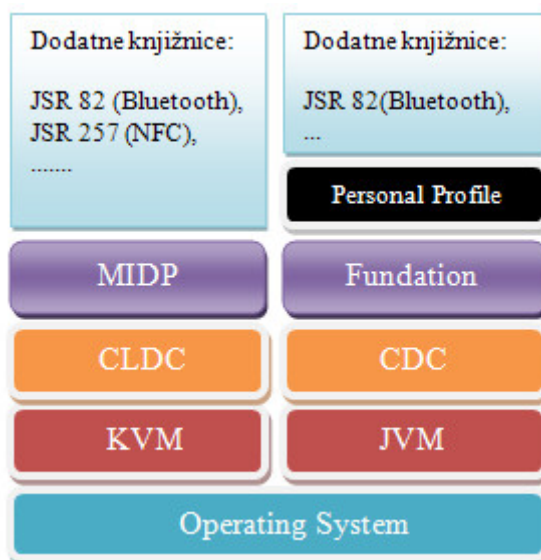
Java ME je tehnologija namenjena razvoju aplikacij za mobilne telefone, dlančnike in ostale mobilne naprave. Je neodvisna od platforme, za interpretacijo njene kode, pa skrbi tolmač ali javanski navidezni stroj JVM ali KVM (ang. Kilobyte Virtual Machine), ki je okrnjena različica JVM navideznega stroja. Razvili so jo pri podjetju Sun Microsystems, njen glavni namen pa je bil prilagoditi Java okolje tako, da bo mogoče izdelovati Java aplikacije, ki se izvajajo na napravah z omejenim pomnilnikom, omejeno velikostjo zaslona in omejeno količino energije.

Platforma Java ME je zbirka tehnologij in specifikacij, z uporabo katerih lahko proizvajalci elektronskih naprav tvorijo JRE, ki je prilagojeno potrebam posameznih naprav. Danes velika večina sodobnih mobilnih naprav (od preprostih mobilnih telefonov, do zmogljivih dlančnikov) nudi ustrezno okolje za poganjanje Java ME aplikacij.

Sestavljena je iz treh osnovnih elementov:

- Osnovna konfiguracija, ki poskrbi za osnovni nabor knjižnic.
- Profili, ki so množica programirljivih vmesnikov API, ki jih vsebujejo samo nekatere naprave.
- Opcijski tehnološko-specifični programirljivi vmesniki, ki jih podpirajo tehnološko-specifične naprave.

Java ME omogoča dve osnovni konfiguraciji CLDC (ang. Connected limited device configuration) in CDC (ang. Connected device configuration). CLDC je konfiguracija za najpreprostejše naprave (npr. mobilni telefoni, manj zmogljivi PDA-ji, osebni organizatorji), ki imajo majhno procesorsko moč, majhen pomnilnik, majhen zaslon ter omejene možnosti povezovanja. Aplikacije se interpretirajo s pomočjo KVM. CDC pa je konfiguracija za zmogljivejše mobilne naprave (npr. zmogljivi dlančniki, internet TV, navigacijski sistemi v avtomobilih, hišna avtomatizacija), ki imajo močnejše procesorje in večji pomnilnik. Programski model je bližji Javi SE, kar omogoča lažji prenos obstoječih aplikacij. Programsko kodo pa interpretira JVM. Poleg osnovne konfiguracije, lastnosti naprave določajo še profili in specifične knjižnice, ki omogočajo izkoriščanje dodatnih zmožnosti naprave. Najpogosteje uporabljeni profil za konfiguracijo CLDC je MIDP (ang. Mobile Information Device Profile), za konfiguracijo CDC pa profil Foundation. Poleg določenih profilov, pa proizvajalci posamezne naprave opremijo še z dodatnimi specifičnimi programskimi knjižnicami (slika 10).



Slika 10. Platforma Java2 ME.

CDC je zasnovan za naprave, ki imajo naslednje lastnosti:

- 32 bitni mikroprocesor,
- Vsaj 2 MB RAM (ang. Random access memory),
- Vsaj 2,5 MB ROM (ang. Read-only memory).

Konfiguracija CDC je podmnožica Java SE, z nekaj dodanimi razredi. Zadnja verzija CDC je 1.1.2 določena s specifikacijo JSR 218.

CLDC je zasnovan za naprave, ki imajo naslednje lastnosti:

- 16 ali 32 bitni procesor s hitrostjo 16 MHz ali več,
- Najmanj 160 KB pomnilnika rezerviranega za CLDC knjižnice in KVM,
- Najmanj 192 KB celotnega pomnilnika, ki je na voljo za Java platformo,
- Nizko porabo energije,
- Možnost priključevanja na določena omrežja (npr. Brezžično omrežje) z omejeno pasovno širino.

Zadnja konfiguracija CLDC, ki je bila potrjena s strani JCP je CLDC verzija 1.1, določena s specifikacijo JSR 139.

Večina mobilnih naprav, med drugimi tudi Nokia 6212 NFC Classic, podpira CLDC konfiguracijo in profil MIDP. S kombinacijo obeh dobimo Java okolje, ki nam omogoča uporabo ustreznih knjižnic ter programirljivih vmesnikov za razvoj Java ME aplikacij.

MIDP profil

Zadnja različica MIDP profila je 3.0 (JSR 271). Je nadgradnja osnovne verzije MIDP profila 1.0 (JSR 37) in profila MIDP 2.1 (JSR 118).

Vključuje naslednje funkcionalnosti:

- Shranjevanje podatkov na lokalni pomnilnik,
- Midlet programom dovoljuje izvajanje in komuniciranje,
- Možnost povezovanja s strežniki,
- Varnost od začetka do konca povezave,
- Vmesnik za razširjeni grafični uporabniški vmesnik (GUI),
- Specifični vmesnik za razvoj iger,
- Podpora multimediji,
- Razširjeno podporo komunikacijskim protokolom,
- OTA (ang. Over The Air) – nalaganje aplikacij,
- Varna povezavo https in rokovanje s kriptografskimi metodami.

Aplikaciji napisani v Javi ME, ki se izvaja na napravah z MIDP profilom pravimo MIDlet. Sestavljena je iz datoteke s končnico .jar in datoteke s končnico .jad. V Jar (ang. Java Archive) so shranjene vse datoteke, ki so potrebne za izvajanje programa in pripadajoči meta podatki. Jad (ang. Java Application Descriptor) pa nosi opis MIDlet aplikacije, ki je zapakirana v datoteki Jar.

Primer datoteke JAD :

```
MIDlet-Version: 1.0.0
MIDlet-Vendor: Midlet Suite Vendor
MIDlet-Jar-URL: JavaME_BT_NFC_Client_1.jar
MicroEdition-Configuration: CLDC-1.1
MicroEdition-Profile: MIDP-2.0
MIDlet-1: JavaME_BT_NFC_Client_1,c:\nfc.png,j2me_bt_nfc_client
MIDlet-Name: JavaME_BT_NFC_Client_1 Midlet Suite
```

Da se bo MIDlet lahko izvajal na mobilni napravi, mora zadoščati naslednjim zahtevam:

- Glavni razred mora biti izpeljan iz razreda javax.microedition.midlet.MIDlet,
- MIDlet mora biti zapakiran v datoteki .jar,
- Datoteka .jar mora biti overjena s strani CLDC predpreverjevalca (ang. Preverifier).

Izdelava Java ME programa poteka v šestih korakih:

1. Korak: Napišemo kodo v razvojnem orodju.
2. Korak: Napisano kodo prevedemo s prevajalnikom.
3. Korak: Predpreverjanje kode z JVM.
4. Korak: Izdelava paketa Java ME (jar in jad datoteki).
5. Korak: Testiranje z emulatorjem.
6. Korak: Namestitev na mobilno napravo.

Za razvoj aplikacij v Javi ME, potrebujemo ustrezno razvojno okolje in posnemovalnik naprav (ang. Emulator). Pri razvoju aplikacije za interaktiven ogled razstav je bilo uporabljeno razvojno okolje Eclipse z dodatkom Eclipse ME, za namen posnemanja naprav pa Java Me Wireless Toolkit in Nokia 6212 NFC SDK. Alternativna možnost je uporaba razvojnega okolja NetBeans Mobility Pack ali razvojnega paketa podjetja Sun Java ME SDK 3.0, ki poleg razvojnega okolja, že vključuje Java ME Wireless Toolkit.

3.2 Eclipse in Eclipse ME

Eclipse je odprtokodno, razširljivo, integrirano razvojno okolje IDE (ang. Integrated Development Environment), ki se uporablja za izdelavo aplikacij v programskih jezikih Java, C, C++, COBOL, Python, PHP in Perl. Na voljo je za različne operacijske sisteme, kot so Windows, Linux, Solaris. Dobre lastnosti razvojnega okolja, so vseh grafični vmesnik (GUI) in hiter razvoj dodatnih komponent, ki bazirajo na vstavkih (ang. Plug-in) [8].

Nabor operacij Eclipse:

- Ustvarjanje, urejanje in iskanje po projektih,
- Pisanje in urejanje programske kode,
- Razhroščevanje (ang. Debuging) programske kode,
- Prevajanje programske kode,
- Izvajanje aplikacij.

Eclipse ME je vstavek za Eclipse, ki omogoča razvoj MIDlet aplikacij z orodjem Eclipse. Poleg tega pa poskrbi za vključitev orodja Wireless Toolkit, s katerim lažje in hitreje razvijamo ter testiramo razvite aplikacije. Bolj se osredotočamo na sam razvoj in ne toliko na nameščanje in testiranje aplikacije na mobilni napravi.

Nabor operacij Eclipse ME:

- Omogoča nastavljanje Wireless toolkit-a,
- Ustvarjanje Java2ME MIDlet projektov in razvoj MIDlet aplikacij,
- Urejanje JAD datotek,
- Avtomatsko preverjanje class datotek,
- Podpora Java ME posnemovalnikom naprav,
- Pakiranje in razvijanje Java ME programske kode.

3.3 Sun Java 2 Micro Edition (J2ME) Wireless Toolkit za CLDC

J2ME Wireless toolkit, je množica orodij in operacij, ki omogočajo izvajanje Java ME aplikacij, namenjenih mobilnim telefonom in drugim elektronskim napravam, ki imajo konfiguracijo CLDC. Z njim lahko razvijamo aplikacije za naprave, ki ustrezajo specifikacijam JSR 185 (JTWI – Java Technology for the Wireless Industry). Sestavljajo ga orodje za prevajanje in izvajanje aplikacij, posnemovalniki mobilnih naprav in razni pripomočki. Zadnja verzija Wireless toolkit je 2.5.2 in zajema naslednje operacije [9].

Nabor operacij Java ME Wireless toolkit:

- Podpis MIDlet programa,
- Upravljanje s certifikati,
- Integrirano posnemanje komunikacije OTA,
- Posnemanje Push registry (omogoča avtomatski zagon midlet programov, ko se sproži neka akcija (npr. alarm)),
- Izvajanje MIDlet aplikacij v posnemovalnikih naprav.

Specifikacije, ki jih podpira J2ME Wireless toolkit:

- | | |
|---|-----------|
| • Mobile Service Architecture | (JSR 248) |
| • Java Technology for the Wireless Industry (JTWI) | (JSR 185) |
| • Connected Limited Device Configuration (CLDC) 1.1 | (JSR 139) |
| • Mobile Information Device Profile (MIDP) 2.0 | (JSR 118) |
| • PDA Optional Packages for the J2ME Platform | (JSR 75) |
| • Java APIs for Bluetooth | (JSR 82) |
| • Mobile Media API (MMAPI) | (JSR 135) |
| • J2ME Web Services Specification | (JSR 172) |
| • Security and Trust Services API for J2ME | (JSR 177) |
| • Location API for J2ME | (JSR 179) |
| • SIP API for J2ME | (JSR 180) |
| • Mobile 3D Graphics API for J2ME | (JSR 184) |
| • Wireless Messaging API (WMA) 2.0 | (JSR 205) |
| • Content Handler API | (JSR 211) |

- Scalable 2D Vector Graphics API for J2ME (JSR 226)
- Payment API (JSR 229)
- Advanced Multimedia Supplements (JSR 234)
- Mobile Internationalization API (JSR 238)
- Java Binding for the OpenGL(R) ES API (JSR 239)

3.4 XAMPP programski paket

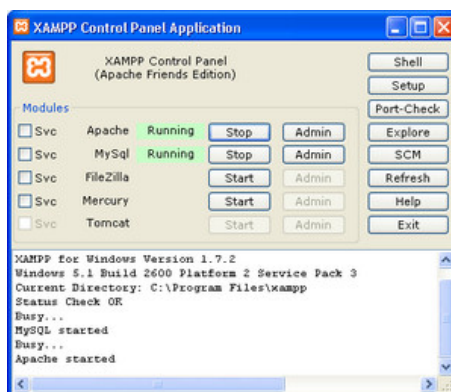
XAMPP (ang. X, Apache, MySQL, PHP and Perl) je brezplačen odprtokoden paket programske opreme, ki skupaj tvori delujoč spletni strežnik. Paket sestavljajo spletni strežnik Apache, sistem za upravljanje s podatkovnimi bazami MySQL in interpreter za programska jezika PHP in Perl. Poleg omenjenih komponent pa XAMPP vsebuje še OpenSSL, ki je odprtokodna implementacija SSL in TLS protokolov in orodje phpMyAdmin. Paket je izdan pod licenco GNU General Public Licence in pogosto nastopa v vlogi spletnega strežnika, ki gostuje dinamične spletne strani. Na voljo je za platforme Microsoft Windows, Linux, Solaris, Mac OS X. Pri razvoju aplikacije, bo Apache uporabljen za gostovanje spletne strani, MySQL za upravljanje s podatkovno bazo, phpMyAdmin pa kot orodje za njeno administracijo.

Namestitev programskega paketa poteka tako, da s spletne strani Softpedia [10] na disk prenesemo izvršljivo datoteko xampp-win32-1.7.2.exe in jo namestimo na računalnik.

Nameščena verzija vsebuje naslednja orodja [11] :

- Apache 2.2.12 (IPv6 enabled) + OpenSSL 0.9.8k,
- MySQL 5.1.37 + PBXT engine,
- PHP 5.3.0,
- PhpMyAdmin 3.2.0.1,
- Webalizer 2.21-02 + GeoIP lite,
- FileZilla FTP Server 0.9.32.

Po namestitvi je uporabniku na voljo XAMPP Control Panel (slika 11), v katerem ročno poženemo strežnika MySQL in Apache. Zagon strežnikov lahko nastavimo tudi tako, da se avtomatsko poženeta ob zagonu operacijskega sistema. MySQL potrebujemo za upravljanje s podatkovnimi bazami, Apache pa za gostovanje spletne strani in za dostop do orodja phpMyAdmin.



Slika 11. Nadzorna plošča XAMPP namenjena zagonu strežnikov Apache in MySQL.

MySQL

MySQL je brezplačna, odprtokodna različica sistema za upravljanje z relacijskimi podatkovnimi bazami. Deluje na različnih platformah kot so Linux, Windows, Mac OS X, HP-UX, AIX, Netware. Za delo s podatki uporablja programski jezik SQL, ki je najbolj razširjen in standardiziran povpraševalni jezik za delo s podatkovnimi zbirkami. Deluje na principu odjemalec/strežnik, pri čemer lahko strežnik namestimo kot sistem, porazdeljen na več strežnikov. Za dostop do podatkovne baze, obstaja veliko število različnih odjemalcev, zbirk ukazov in programskih vmesnikov [12].

PhpMyAdmin

PhpMyAdmin je brezplačno programsko orodje, napisano v programskem jeziku PHP. Namenjeno je administraciji podatkovne zbirke MySQL preko svetovnega spleta. Značilnosti orodja so pregleden grafični vmesnik (GUI), podpora veliko različnih jezikov in širok nabor operacij za delo s podatkovnimi bazami [13].

Nabor operacij phpMyAdmin:

- Podpora večini MySQL operacij:
 - Urejanje podatkovnih baz, tabel, relacij, polj in indeksov,
 - Kreiranje, kopiranje, brisanje, preimenovanje in spreminjanje podatkovnih baz, tabel, polj in indeksov,
 - Vzdrževanje strežnika, podatkovnih baz in tabel,
 - Izvajanje, urejanje in knjiženje SQL stavkov,
 - Upravljanje z uporabniki in njihovimi privilegiji,
 - Upravljanje s shranjenimi (store) procedurami in prožilci.
- Uvoz podatkov iz različnih formatov in izvoz podatkov v različne formate,
- Kreiranje grafičnega prikaza ureditve podatkovnih baz.

3.5 Nokia PC Suite

Nokia PC suite je programski paket, ki omogoča izmenjavo podatkov med mobilnim telefonom in osebnim računalnikom. Običajno ga dobimo ob nakupu mobilnega naprave, lahko pa si ga brezplačno prenesemo s spletne strani Nokia PC Suite. Ko si paket prenesemo, ga namestimo na osebni računalnik, vzpostavimo povezavo, nato pa izbiramo med operacijami, ki jih ponuja [14].

Nabor operacij paketa Nokia PC Suite:

- Varnostna kopija podatkov (prenesemo podatke iz telefona na računalnik ali obratno),
- Sinhronizacija osebnih podatkov med telefonom in računalnikom,
- Telefon uporabimo kot modem, za povezavo računalnika z internetom,
- Urejanje, ustvarjanje in ogledovanje vizitk shranjenih v telefonu,
- Pošiljanje besedilnih in večpredstavnostnih sporočil iz računalnika,
- Urejanje ter vnos dogodkov na koledar v telefonu,
- Prenašanje datotek (slike, avdio, video) med telefonom in računalnikom,

- Nameščanje programov in iger iz računalnika na telefon,
- Posodabljanje programske opreme na telefonu,
- Prenašanje zemljevidov iz interneta, ki jih uporabljamo v aplikaciji zemljevidi Nokia.

3.6 Series 40 5. edition Nokia 6212 NFC SDK

Nokia 6212 NFC SDK je paket programske opreme za razvoj aplikacij, namenjenih mobilnemu telefonu Nokia 6212 Classic NFC. Osnovan je na paketu Series 40 izdaja 5, ki mu je dodan paket za razvoj NFC aplikacij. Sestavljen je iz razvojnega in testnega okolja za aplikacije, ki uporabljajo vmesnik za brezkontaktno komunikacijo (JSR-257), MMS, SMS, WAP in XHTML brskalnik. Vključuje tudi grafičnega vmesnik, ki posnema izgled in funkcionalnosti naprave Nokia 6212 Classic NFC in simulator RFID značk. Možna je integracija paketa v razvojno okolje Eclipse 3.3.1 z dodatkom Eclipse ME 1.7.6 in Netbeans Mobility 6.0.

NFC paket podpira:

- Čitalce OMNIKEY in Pegoda,
- Značke Mifare Ultralight, 1k, 4k ter značke skladne s standardom ISO 14443-4.

Specifikacije, ki jih podpira Nokia 6212 NFC SDK:

- JSR-257 Contactless Communication API,
- JSR-118 Mobile Information Device Profile 2.1,
- JSR-118 Mobile Information Device Profile 2.0,
- JSR-135 Mobile Media API 1.2,
- JSR-139 Connected Limited Device Configuration 1.1,
- JSR-177 APDU and CRYPTO optional packages of the Security and Trust Services API for J2ME(TM) 1.0,
- JSR-75 FC and PIM optional packages of the PDA Optional Packages for the J2ME™ Platform,
- JSR-82 Java(TM) API for Bluetooth 1.1,
- JSR-82 Java(TM) API BT Obex,
- JSR-120 Wireless Messaging API 1.1 for SMS,
- JSR-172 Web Services JAXP, XML-RPC API 1.0 of the J2ME™ Web Services Specification,
- JSR-184 Mobile 3D Graphics API for J2ME(TM) 1.1,
- JSR-185 Java(TM) Technology for Wireless Industry 1.0,
- JSR-205 Wireless Messaging API 2.0 for SMS and MMS,
- JSR-211 Content Handler API,
- JSR-226 Scalable 2D Vector Graphics API for J2ME(TM) 1.1,
- JSR-234 Advanced Multimedia Supplements 1.0,
- JSR-248 Mobile Service Architecture (Subset) 1.0,
- Nokia UI API 1.1.

3.7 Bluesoleil

Bluesoleil je licenčen programski paket, ki ga je leta 2002 razvilo podjetje IVT Corporation. Vsebuje gonilnike in programsko opremo za nadzor Bluetooth naprave, priključene na osebni ali prenosni računalnik. Omogoča Bluetooth komunikacijo s široko množico naprav (slika 12), kot so mobilni telefoni, stereo/mono slušalke, tipkovnice, miške, kamere, tiskalniki, GPS naprave in dlančnike. Na voljo je za operacijske sisteme Windows XP, Vista, Windows 7, Linux, WinCE in Windows Mobile [15].

Programska oprema omogoča:

- Podpora VoIP telefoniji,
- Pregled in upravljanje kontaktov shranjenih na mobilni napravi,
- Pošiljanje, kopiranje SMS sporočil preko računalnika,
- Podpira profila A2DP in AVRCP,
- Prenasjanje datotek iz računalnik na mobilno napravo in obratno,
- Možnost tvorjenja PAN (ang. Personal area network) omrežij,
- Tiskanje datotek.



Slika 12. Možnosti uporabe Bluesoleil + Bluetooth USB Adapterja.

(Vir: Internet: http://www.bluesoleil.com/products/images/main_scenarios.gif)

4. Strojna oprema

To poglavje je namenjeno predstavitvi strojne opreme, ki je bila uporabljena pri razvoju in testiranju aplikacij.

Za razvoj in testiranje aplikacije v realnem okolju je bil uporabljen osebni računalnik, ki ima vzpostavljeno brezžično Wi-Fi ali Bluetooth omrežje, mobilna naprava, ki zadosti spodnjim zahtevam in z izbrano mobilno napravo kompatibilne RFID značke.

- Podpora NFC komunikaciji,
- Podpora vsaj še ene izmed ostalih obstoječih brezžičnih tehnologij (Bluetooth, Wi-Fi),
- Podpora jezika Java in ustreznih aplikacijsko programirljivih vmesnikov, za razvoj aplikacije.

4.1 Osebni računalnik z Bluetooth modulom

Osebni računalnik z Bluetooth modulom (slika 13), ki je bil uporabljen pri razvoju programske opreme in testiranju aplikacij.



Tehnične lastnosti PC:

- procesor: Intel(R) Core™2 Duo CPU E4500 @ 2.20 GHz,
- osnovna polšča: Asus P5LP-LE,
- grafična kartica: NVIDIA GeForce 8600 GT (512 MB),
- pomnilnik: 2048 MB DDR2,
- trdi disk: Hitachi 250 GB,
- monitor: NEC LCD 22WV,
- CD-DVDRW TS-H653N,
- Bluetooth USB Adapter CNR-BTU5.

Slika 13. Desktop PC + USB Bluetooth adapter.

Programska oprema:

- Microsoft Windows XP Professional + service pack 3,
- Eclipse + Eclipse ME,
- XAMPP (Apache, MySQL, PHP, Perl),
- Nokia developer tools series 40 (za Nokia 6212),
- Sun Java jdk 1.6.0_13,
- Nokia PC Suite,
- Bluesoleil 2.7.0.35.

Tehnične lastnosti Bluetooth USB Adapter CNR-BTU5:

- Bluetooth v2.0 + EDR (Enhanced Data Rate),
- Hitrost prenosa podatkov do 3 Mbps,
- Domet do 10 m,
- Združljiv z USB 2.0 napravami,
- Velikost 22 x 14 x 5 mm.

4.2 NFC mobilni telefon

Mobilni telefon, ki je bil uporabljen pri testiranju aplikacij je Nokia 6212 Classic (slika 14). To je zanesljiv in intuitiven mobilni telefon, ki združuje klasičen dizajn ter preverjeno tehnologijo. Nudi podporo komunikaciji v bližnjem polju (NFC) z možnostmi branja, pisanja in deljenja [14].



Tehnične lastnosti:

Glavne lastnosti:

- Ločljivost zaslona: 240 x 320 pik ,
- Dimenzije: 114,7 x 47,1 x 14,5 mm,
- Teža : 88 g,
- Fotoaparati: 2 milijona pik, bliskavica, 8x povečava,
- Visoko hitrostna 3G povezava,
- Koledar, seznam opravkov, beležka,
- Predvajalnik glasbe in radio,
- Pomnilnik: 22 MB + reža za micro SD kartico,
- Operacijski sistem: Nokia OS,
- FOTA Firmware over the Air,
- Delovna frekvenca: UMTS 850/2100
GSM/EDGE 850/900/1800/1900,

Povezovanje:

- Komunikacija v bližnjem polju (NFC) z možnostmi branja, pisanja in deljenja,
- Vmesnik Bluetooth 2.0 + EDR,
- 2,5 mm AV vtič,
- Vmesnik micro USB 2.0,
- GPRS: multislot razreda 10,
- EDGE: multislot razreda 10,
- Polno oskrbovanje odjemalcev OMA,
- Bluetooth profili : A2DP, AVRCP, DUN, FTP, GAP, GAVDP, GOEP, HFP, HSP, OPP, SAP, SDAP, SPP.

Slika 14. Nokia 6212 Classic.

Avdio in video:

- Avdio formati: AAC, AAC+, AMR-NB, AMR-WB, eAAC+, MIDI Tones (poly 64), Mobile XMF, MP3, MP4, SP-MIDI, True tones, WMA,
- Video formati: 3GPP formati(H.263), H.264/AVC,MPEG-4,
- Slikovni formati: BMP, GIF87a, GIF89a, JPEG, PNG, WBMP.

Razvojna platforma:

- Serija 40, 5. Izdaja.

Sporočanje:

- Sporočila: IM, MMS+SMIL, SMS,
- Protokoli za elektronsko pošto: IMAP4, POP3, SMTP.

Brskalnik in Flash:

- Brskalnik: Opera Mini, WAP 2.0, XHTML over TCP/IP,
- Flash Lite 2.1.

Vmesniki ki jih podpira:

- JSR 139 Connected, Limited Device Configuration (CLDC) 1.1,
- JSR 118 MIDP 2.1,
- JSR 248 Mobile Service Architecture for CLDC,
- JSR 75 FileConnection and PIM API,
- JSR 82 Java™ APIs for Bluetooth 1.1,
- JSR 135 Mobile Media API 1.1,
- JSR 172 J2ME™ Web Services Specification (RPC package),
- JSR 172 J2ME™ Web Services Specification (XML Parser package),
- JSR 177 Security and Trust Services API za J2ME™ (SATSA-APDU package),
- JSR 177 Security and Trust Services API za J2ME™ (SATSA-CRYPTO package),
- JSR 184 Mobile 3D Graphics API for J2ME™,
- JSR 205 Wireless Messaging API 2.0,
- JSR 211 Content Handler API,
- JSR 226 Scalable 2D Vector Graphics API,
- JSR 234 Advanced Multimedia Supplements 1.1 (audio3d),
- JSR 234 Advanced Multimedia Supplements 1.1 (music),
- JSR 257 Contactless Communication API,
- JSR 257 Extensions for Nokia 6212 classic,
- Nokia UI API 1.1.

Oprema za razvoj aplikacij SDK:

- Series 40 Nokia 6212 NFC SDK (omogoča razvoj Java™ aplikacij, z možnostjo uporabe NFC za mobilni telefon Nokia 6212 classic).

- Series 40 Platform SDKs (omogoča razvijalcem hitro in učinkovito testiranje in razvoj Java in Flash Lite aplikacij. SDK serija 40 6.ta izdaja, vsebuje podporo za JME™ (JSR-179), Flash Lite 3 od Adobe, in WebKit odprtokodni brskalnik.

Ker je tehnologija NFC razmeroma nova, na Slovenskem trgu v prosti prodaji nisem zasledil NFC naprav ali NFC telefona. Stanje NFC mobilnih telefonov na svetovnem trgu, pa je opisano v naslednjem odstavku.

Prva NFC naprava je na trg prišla leta 2004. To je bil mobilni telefon Nokia 3220 z NFC modulom. Naslednji mobilni telefon, ki je omogočal NFC komunikacijo je bila Nokia 6131. Omenjena naprava je v celoti podpirala specifikacije NFC foruma, uporabnikom pa so poleg telefona, brezplačno ponudili tudi zbirko že napisanih knjižnic (SDK) in posnemovalnik naprave. Zbirka programerjem omogoča lasten razvoj NFC aplikacij, namen posnemovalnika pa je lažje testiranje in razhroščevanje razvijajočih aplikacij, saj jih ni potrebno vsakokrat fizično nameščati na mobilno napravo.

Poleg Nokie, ki je nekako pionir na področju NFC naprav so na trg NFC mobilne telefone lansirali še Benq (model T80), Samsung (SGH x700 NFC, D500E), Sagem (my 700X Contactless), LG (600V Contactless) in Motorola (L7 (SLVR)). Zadnji napravi, ki jih je na trg poslala Nokia sta modela 6212 Classic in 6216 Classic. Obe napravi imata polno podporo NFC tehnologiji in tudi ustrezna paketa za razvoj aplikacij z ustreznim posnemovalnikom naprave. Ravno zaradi široke podpore in ustreznega paketa za razvoj aplikacij v programskem jeziku Java, je bila za testiranje aplikacije uporabljen telefon Nokia 6212 Classic.

Nokia 6212 Classic podpira NFC tehnologijo in Bluetooth. Ne podpira pa Wi-Fi, zato se bodo podatki med odjemalcem in strežnikom prenašali preko brezžične tehnologije Bluetooth, in ne preko Wi-Fi, kar bi morda bilo bolje, saj so Wi-Fi omrežja v praksi veliko bolj razširjena.

4.3 RFID značke, za telefon NOKIA 6212 Classic NFC

NFC čitalec v mobilnem telefonu Nokia 6212 Classic NFC, je združljiv s pasivnimi brezkontaktnimi značkami (energijo potrebno za delovanje dobijo neposredno od signala, ki se inducira v anteni), ki so narejene po standardu ISO/IEC 14443.

ISO 14443 standard predpisuje način komunikacije in prenosni protokol med pametno kartico in čitalnikom. Sestavljen je iz štirih delov in je predpisan za dva tipa kartic: Tip A in Tip B. Oba tipa komunicirata preko radijskih valov, pri frekvenci 13.56 MHz in za prenos uporabljata enak komunikacijski protokol, opisan v četrtem delu standarda ISO 14443-4. Razlikujeta pa se v implementaciji drugega ISO 14443-2 in tretjega dela standarda ISO 14443-3.

Prvi del standarda predpisuje fizične lastnosti in karakteristike brezkontaktnih značk. Drugi del predpisuje lastnosti signala in moč radijskih frekvenc. V tretjem delu je predpisan postopek uvodnih nastavitvev oz. inicializacije in protikolizijski protokol. Četrty del pa predpisuje prenosni protokol.

Tipi značk, ki jih lahko uporabljamo z mobilnim telefonom Nokia 6212 Classic NFC: [14]

- MIFARE Classic 1k , 4k,
- MIFARE Ultralight,
- MIFARE DESFire ,
- Sony FeliCa (nezaščitene),
- Innovision Jewel (samo za branje) in Topaz,
- Ostale značke, ki so narejene v skladu s standardom ISO 14443.

MIFARE je vodilni industrijski standard oziroma tehnologija, ki se uporablja v sistemih, ki jih sestavljata pametna značka in čitalec. Osnovana je na podlagi protokola ISO 14443 tip A. Razvili so jo v družbi NXP Semiconductors, ki je bila leta 2006 ustanovljena s strani podjetja Philips. Je ena najpogostejše uporabljajih tehnologij, ki se jih vgrajuje v brezkontaktno značke in čitalce brezkontaktnih značk [16].

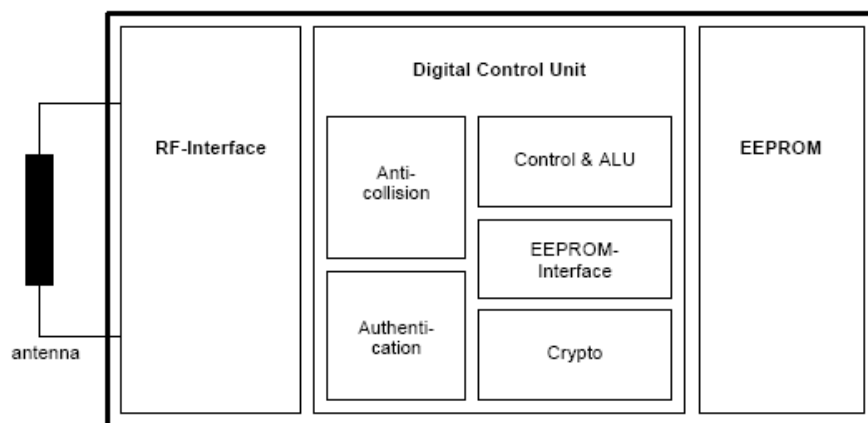
MIFARE Classic (Standard) so značke v osnovi namenjene shranjevanju podatkov, ki niso zaupne narave. Za zaščito uporabljajo 48 bitni ključ, ki pa danes ni več zanesljiv. Velikost pomnilnika v značkah je 1k oziroma 1024 bajtov (16 sektorjev). Na voljo je tudi različica, ki ima pomnilnik velik 4k ali 4096 bajtov (40 sektorjev). Lahko se jih uporablja za branje in pisanje. Prvih 16 bajtov pomnilnika je namenjeno serijski številki značke UID (ang. Unique Identification Number) in podatkom o izdelovalcu. Nato pa je v vsakem sektorju 16 bajtov rezerviranih za ključ in pravice dostopa. Ti deli pomnilnika ne morejo biti uporabljeni za shranjevanje podatkov, tako da je dejanski pomnilnik, ki je na voljo za shranjevanje podatkov pri 1k karticah 752 bajtov, pri 4k karticah pa 3440 bajtov. Ker so bile značke MIFARE Classic enostavne in cenovno dostopne, so se veliko uporabljale v povezavi s transportom, elektronskimi denarnicami, vstopnicami ter pri kontroli dostopa. Danes jih zamenjujejo značke MIFARE Plus, ki za zaščito uporabljajo 128 bitno AES kodiranje.

MIFARE (Ultralight) so nizkocenovne značke, ki za delovanje uporabljajo enak protokol, kot MIFARE Classic. Od njih se razlikujejo po tem, da ne vsebujejo varnostne zaščite in imajo za odtенок različne ukaze. Na voljo imajo 512 bitov pomnilnika, v praksi pa se pogosto uporabljajo kot vstopnice na prireditvah.

MIFARE (DESFire) so značke, ki za delovanje uporabljajo NXP mikroprocesor in naprednejše varnostne mehanizme. Na trgu so dostopne 4 različice: prva za zaščito uporablja trojni DES šifrirni algoritem in ima na voljo 4 k prostora. Ostale tri različice, za zaščito uporabljajo AES šifrirni algoritem in imajo na voljo 2k, 4k ali 8k pomnilnika.

Za hrambo zaupnih podatkov so bolj primerne značke MIFARE Desfire ev1 in MIFARE Plus, saj uporabljajo naprednejše mehanizme zaščite[17].

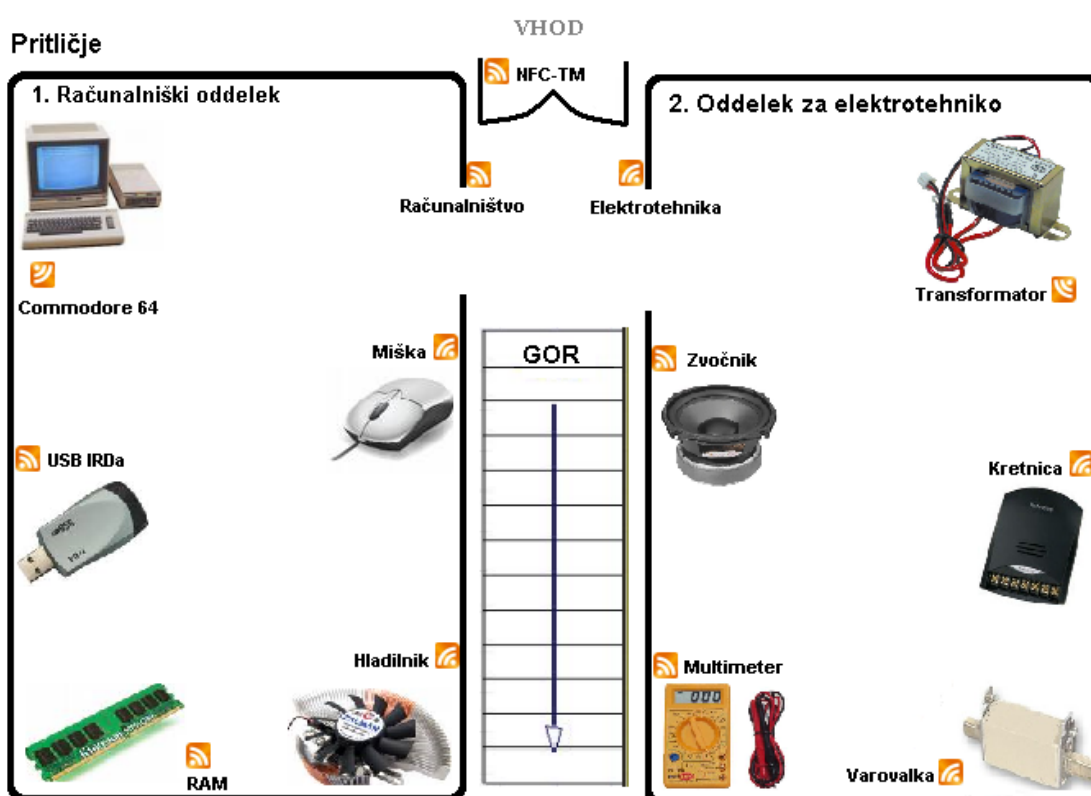
V sklopu diplomske naloge, so bile za označevanje razstavnih eksponatov, uporabljene značke MIFARE Classic (slika 15). Značka je sestavljena iz EEPROM pomnilnika (1Kb), vmesnika RF in digitalno kontrolne enote. Energija za prenos in podatki se prenašajo preko antene, ki je sestavljena iz nekaj ovojev žice.



Slika 15. Block diagram značke Mifare Classic
(Vir: <http://www.nxp.com>)

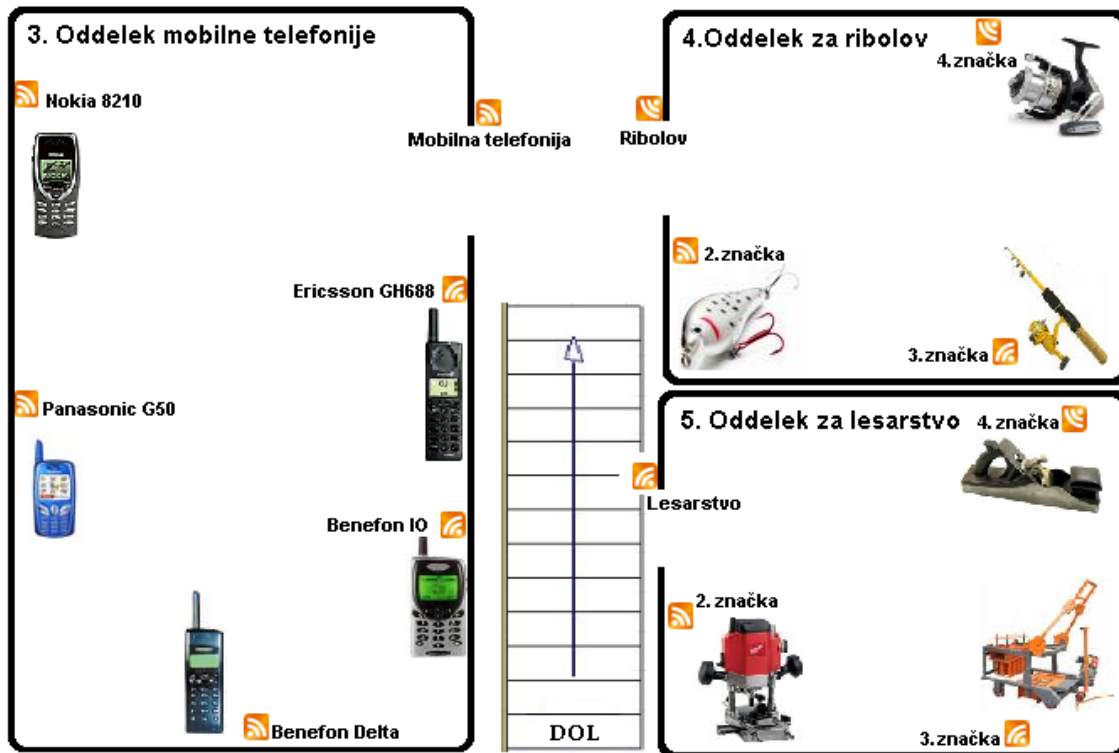
5. Razvoj aplikacije za interaktiven ogled razstave

Za razvoj aplikacije je uporabljen poenostavljen primer tehniškega muzeja z imenom NFC-TM, ki je razdeljen na pritličje in prvo nadstropje. Pritličje (slika 16) vključuje oddelek za računalništvo in oddelek za elektrotehniko, v prvo nadstropje (slika 17) pa spadajo oddelek za mobilno telefonijo, ribolov ter oddelka za lesarstvo. Na vhodu muzeja je RFID značka, ki obiskovalcu posreduje osnovne informacije o muzeju in informacije o lokaciji ter stanju posameznih oddelkov. Naslednje RFID značke so pred vhodi v posamezni oddelek. Te značke obiskovalcu posredujejo opis oddelka in osnovne informacije o razstavnih eksponatih, ki jih na posameznem oddelku najdemo. V oddelkih so razstavljeni razstavni eksponati, ki so prav tako opremljeni z RFID značkami. Te značke omogočajo uporabniku posredovanje informacije in slike o razstavnih eksponatih.



Slika 16. Pritličje muzeja NFC-TM.

1. nadstropje



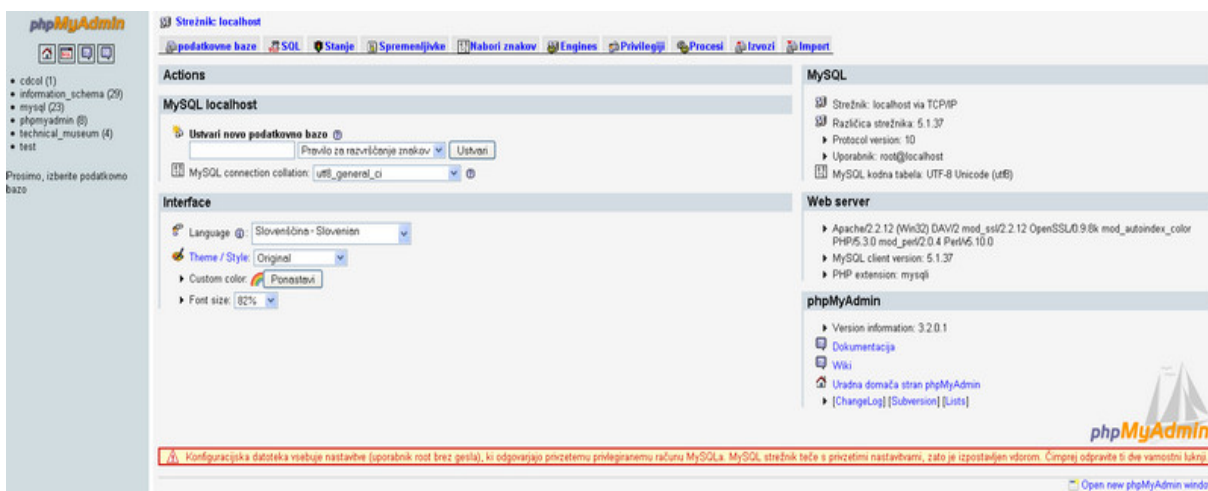
Slika 17. Prvo nadstropje muzeja NFC-TM.

Praktični del diplomske naloge obsega razvoj aplikacij:

- Razvoj aplikacije za PC, ki bo delovala kot strežnik. Namen aplikacije je, da odjemalcem preko Bluetooth povezave posreduje ustrezne podatke iz podatkovne baze tehničnega muzeja (ang. technical museum), hkrati pa vse poizvedbe zapisuje v tabelo statistika (ang. statistics), ki se nahaja v isti podatkovni bazi.
- Razvoj aplikacije za mobilni telefon Nokia 6212 Classic, ki z RFID značke prebere unikatno identifikacijsko številko, jo preko Bluetooth povezave pošlje strežniku in počaka odgovor. Ko strežnik odgovori z opisom in sliko razstavnega eksponata, aplikacija na mobilnem telefonu poskrbi za prikaz podatkov na zaslonu.
- Razvoj aplikacije za PC, ki služi kot grafični vmesnik, za vpis oddelkov in razstavnih eksponatov, pregled in nastavljanje statusa posameznim oddelkom ter za izračun in pregled statistike. V okviru statistike se vodi podatke o obiskovalcih in njihovem številu, ogledanih razstavnih eksponatih in njihovi priljubljenosti.
- Razvoj spletne strani, ki iz podatkovne baze tehničnega muzeja prikaže podatke o oddelkih in razstavnih eksponatih, ki se v njih nahajajo.

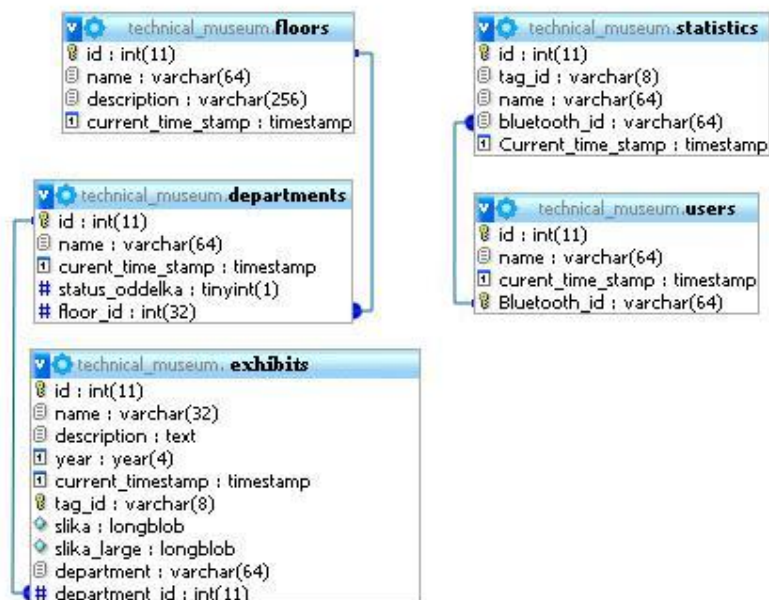
5.1 Podatkovna baza

Za hranjenje informacij, je bilo potrebno na osebni računalnik namestiti ustrezno podatkovno strukturo. Uporabljen je bil programski paket XAMPP, ki vsebuje sistem za upravljanje s podatkovnimi bazami MySQL in orodje za njihovo administracijo phpMyAdmin. Po zagonu strežnika MySQL in Apache, do orodja phpMYadmin (slika 18), dostopamo tako, da v spletni brskalnik vnesemo naslov: <http://localhost/phpmyadmin/>.



Slika 18. Orodje za administracijo podatkovnih baz MySQL.

Pri razvoju aplikacije, je bila ustvarjena podatkovna baza z imenom 'tehnični muzej' (ang. technical museum). Definirane so bile ustrezne tabele in relacije med njimi. Struktura, relacije in tipi podatkov, ki sestavljajo posamezne tabele, so prikazani na sliki 19.



Slika 19. Struktura tabel v podatkovni bazi tehničnega muzeja.

Podatkovno bazo 'tehnični muzej' sestavlja pet tabel. Tabela 'nadstropja' (ang. floors) hrani podatke o nadstropjih. V našem primeru, sta to pritličje in 1.nadstropje. Tabela 'oddelki' (ang. departments), hrani podatke o oddelkih in njihovem statusu (odprt, zaprt). Tabela 'razstavnih eksponati' (ang. exhibits), ki je prikazana na sliki 20, hrani osnovne podatke in slike o razstavnih eksponatih. Tabela 'statistika' (ang. statistics), hrani podatke o obiskovalcih in obiskanih razstavnih eksponatih. Na podlagi te tabele se oblikujejo statistični izračuni. Tabela 'uporabniki' (ang. users) pa vsebuje podatke o obiskovalcih muzeja.

id	name	description	year	current_timestamp	tag_id	slika	department	dep_id
1	Commodore 64	Commodore 64 je bil zelo popularen računalnik v 80...	1982	2010-01-04 17:21:50	3c3d5425	[BLOB - 7,1 KB]	computer_department	1
2	USB IrDa	Chip U2-Irda wireless infrared enables infrared on...	2002	2010-01-04 17:14:18	bc235925	[BLOB - 2,8 KB]	computer_department	1
3	RAM Pomnilnik	Bralno-pisalni pomnilnik, pogosto imenovan z angle...	1990	2010-01-04 17:14:23	9c9f5825	[BLOB - 3,8 KB]	computer_department	1
4	Hladilnik CPU	Computer cooling is the process of removing heat f...	1986	2010-01-11 17:01:18	8c25425	[BLOB - 4,2 KB]	computer_department	1
5	Racunalniška miška	In computing, a mouse (plural mouses, mice, or mou...	1981	2010-01-12 14:29:33	9ca75825	[BLOB - 3,1 KB]	computer_department	1
9	Zvočnik	A loudspeaker (or "speaker") is an electroacoustic...	1950	2010-01-12 14:25:45	7cc55825	[BLOB - 4,4 KB]	electrical_engineering_department	2
10	Varovalka	In electronics and electrical engineering a fuse (...)	1955	2010-01-04 17:14:46	fd35425	[BLOB - 2,0 KB]	electrical_engineering_department	2
17	Transformator	Transformator je električna naprava, ki pretvarja ...	1957	2010-01-04 17:14:57	3c9f5825	[BLOB - 8,7 KB]	electrical_engineering_department	2

Slika 20. Tabela razstavnih eksponatov.

Vse tabele vsebujejo polje z unikatno identifikacijsko številko (ang. id) in čas, ko je bil posamezen vnos dodan v tabelo (ang. Current time stamp). Poleg omenjenih polj, tabela nadstropja vsebuje še imena nadstropij in njihov opis. Tabela oddelkov vsebuje še imena oddelkov, njihov status in identifikacijsko številko nadstropja v katerem se oddelek nahaja. Tabela razstavnih eksponatov vsebuje še identifikacijsko številko oddelka v katerega sodi razstavni eksponat, identifikacijsko številko RFID značke ter imena, opis, leto izida in sliko razstavnih eksponatov. Tabela statistika, pa poleg polja za identifikacijsko številko in trenutnega časa, vsebuje še številko RFID značke razstavnega eksponata in Bluetooth identifikacijsko številko uporabnika, ki si je razstavni eksponat ogledal s pomočjo tehnologije NFC.

5.2 Razvoj Bluetooth strežnika za PC

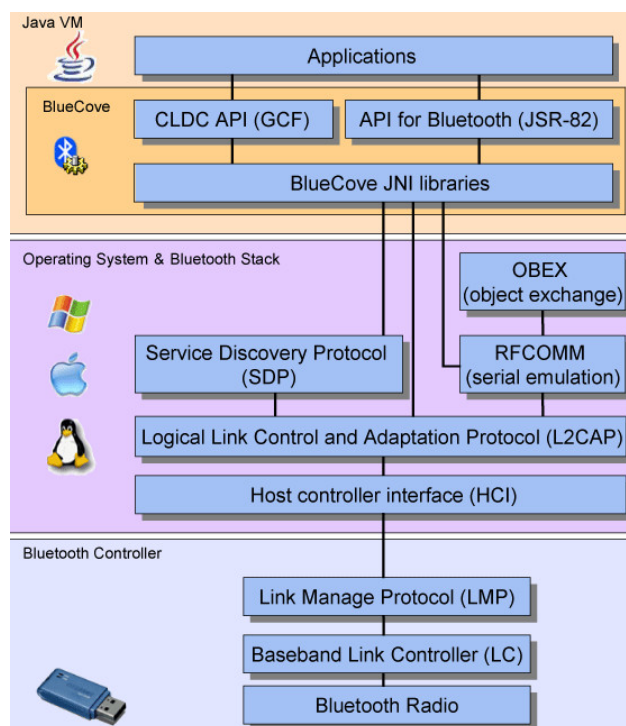
Za osebni računalnik je bilo potrebno razviti aplikacijo v Javi SE, ki bo delovala kot Bluetooth strežnik. Ko se bo odjemalec povezal, bo strežnik ustvaril novo nit, v kateri bo potekala medsebojna komunikacija, hkrati pa bo strežnik čakal na drugega odjemalca. Komunikacija bo delovala tako, da odjemalec pošilja unikatne identifikacijske številke razstavnih eksponatov. Strežnik bo s pomočjo te številke naredil poizvedbo v podatkovni bazi in odjemalcu posredoval sliko in opis razstavnega eksponata. Poleg omenjenega pa bo naloga strežnika še zapis vsake poizvedbe v tabelo statistika. Na podlagi podatkov v tej tabeli, se bo vodila statistika priljubljenosti oddelkov in razstavnih eksponatov.

5.2.1 Bluetooth strežnik

Pri izdelavi aplikacije, ki deluje kot Bluetooth strežnik, je bila uporabljena knjižnica Java razredov Bluecove, ki implementira vmesnik JSR 82. Umestitev Bluecove v Bluetooth sklad je prikazana na sliki 21. Knjižnico je možno uporabljati skupaj z Javo SE od različice 1.1 naprej. Uporablja se jo lahko v navezavi z Bluetooth skladi WIDCOMM, BlueSoleil in

Bluetooth skladi, ki so del operacijskih sistemov Mac OS X, Microsoft XP SP2, Vista in Windows Mobile [18]. V njej so podprti naslednji Bluetooth profili:

- SDAP – Service Device Application Profile,
- RFCOMM – Serial Cable Emulation Protocol,
- L2CAP – Logical Link Control and Adaptation Protocol,
- OBEX – Generic Object Exchange Profile (GEOP).



Slika 21. Umestitev Bluecove knjižnice v Bluetooth sklad.

(Vir: Internet: <http://bluecove.org/images/stack-diagram.png>)

JSR 82 je uraden Java vmesnik namenjen izdelavi aplikacij, ki za komunikacijo uporabljajo komunikacijo Bluetooth. Z uporabo vmesnika lahko implementiramo aplikacije, ki vsebujejo naslednje funkcije:

- Določanje in pregledovanje lastnosti lokalne Bluetooth naprave,
- Vzpostavitev RFCOMM, L2CAP in OBEX komunikacije med napravami,
- Odkrivanje oddaljenih naprav in njihovih servisov v dometu naše naprave,
- Izdelava aplikacij, ki delujejo kot odjemalci v komunikaciji s strežniki,
- Izdelava aplikacij, ki delujejo kot strežniki v komunikaciji z odjemalci,
- Zagotavljanje varnosti pri komunikaciji.

Za prenos podatkov med Bluetooth napravami, je bilo potrebno izbrati med tremi protokoli OBEX, RFCOMM in L2CAP. V aplikaciji je uporabljen Serial Port Profil z uporabo RFCOMM protokola.

Osnovni koncept Bluetooth aplikacije, ki ima vlogo strežnika je sestavljen iz naslednjih komponent:

1. Konstruiranje URL naslova, ki nosi informacijo o tem, kako se odjemalec lahko poveže,
2. Odpiranje povezave, na katero se odjemalci lahko priključijo,
3. Vzpostavitev povezave z odjemalcem,
4. Sprejemanje in pošiljanje podatkov med strežnikom in odjemalcem.

Konstruiranje URL naslova za povezovanje z odjemalcem

Vsaka Bluetooth aplikacija potrebuje univerzalen identifikator UUID (ang. Universal Unique Identifier), za preverjanje delujočih aplikacij na strežniku in na odjemalcu in ime Bluetooth servisa. UUID dobimo tako, da ustvarimo objekt razreda UUID, ime servisa pa definiramo s podatkovnim tipom String.

```
UUID uuid = new UUID("32 ali 128 bit identifikacijska številka");
String ime_servisa = "Servis_1";
```

Iz zgornjih dveh spremenljivk je potrebno tvoriti URL naslov, ki se ga uporabi za kreiranje Bluetooth konektorja.

Za naslov URL je podan zapis: "btspp://localhost:" + uuid + ";name = ime_servisa; authorize = false". Opis parametrov URL naslova je v tabali 4.

Parameter	Opis
btspp	Predstavlja uporabljeni protokol, ki je v našem primeru RFCOMM,
UUID	Predstavlja univerzalno unikatno identifikacijsko številko,
name	Predstavlja ime Bluetooth servisa,
authorize	Nam pove ali je oddaljena (ang. remote) naprava pooblaščen, da lahko odpre Bluetooth povezavo.

Tabela 4. Parametri naslova URL.

Poleg konstruiranja URL naslova, moramo za uspešno postavitve strežnika, lokalno Bluetooth napravo pripraviti za delovanje. To storimo tako, da ustvarimo objekt razreda *LocalDevice* in pokličemo metodo *setDiscoverable*.

```
local = LocalDevice.getLocalDevice();
String BT_address = local.getBluetoothAddress(); //vrne BT naslov lokalne BT naprave
String BT_name = local.getFriendlyName(); // vrne ime lokalne Bluetooth naprave

if (!local.setDiscoverable(DiscoveryAgent.GIAC)) {
    System.out.println("Spodletel poizkus. Lokalna BT naprava ni vidna!");
}
```

Local Device in *Remote Device* sta glavna razreda, namenjena vodenju podatkov o Bluetooth napravah. *Local device* je razred za vodenje podatkov o lokalni bluetooth napravi (PC), *Remote device* pa razred za vodenje podatkov o oddaljeni napravi (mobilni telefon).

Statična metoda *LocalDevice.getLocalDevice* vrne pojavitev (ang. Instance) objekta *LocalDevice*. S klicem metode *local.setDiscoverable(DiscoveryAgent.GIAC)*, pa naredimo našo Bluetooth napravo vidno ostalim Bluetooth napravam.

Odpiranje povezave, na katero se odjemalci lahko priključijo

Za odpiranje povezave uporabimo konektor, ki ga ustvarimo z razredom *StreamConnectionNotifier* in razredom *StreamConnection*. Oba razreda prihajata iz paketa *javax.microedition.io*, ki je v našem primeru vsebovan v knjižnici Bluecove. Preden povezavo odpremo je potrebno registrirati Bluetooth servis. Registracija servisa pomeni ustvariti zapis, ki ga oddaljene Bluetooth naprave lahko poiščejo. To storimo s klicem.

```
StreamConnectionNotifier notifier = null;
notifier = (StreamConnectionNotifier) Connector.open(url);
```

Ko je zapis servisa narejen povezavo, na katero se odjemalci lahko povežejo, odpremo s klicem metode *acceptAndOpen()*. Ta metoda čaka na odjemalca in aplikacijo blokira do takrat, ko odjemalec pošlje ustrezno zahtevo za vzpostavitev povezave s strežnikom.

```
StreamConnection conn = null;
StreamConnection conn = notifier.acceptAndOpen();
```

Vzpostavitev povezave z odjemalcem

Ko strežnik prejme zahtevo za vzpostavitev povezave, metoda *acceptAndOpen()* vrne objekt razreda *StreamConnection* s katerim se upravlja tokove med lokalno in oddaljeno Bluetooth napravo. S klicem metode *openOutputStream()*, ustvarimo izhodni tok, s katerim pošljamo sporočila na odjemalčevo Bluetooth napravo. S klicem metode *openInputStream()*, pa ustvarimo vhodni tok, s katerim sprejemamo sporočila iz odjemalčeve Bluetooth naprave.

Naslov oddaljene naprave dobimo s klicem metode *RemoteDevice.getBluetoothAdress()*.

```
remote = RemoteDevice.getRemoteDevice(conn);
String remoteAddress = remote.getBluetoothAdress();
```

Sprejemanje in pošiljanje podatkov med strežnikom in odjemalcem

Podatki med odjemalcem in strežnikom se prenašajo preko tokov. Komunikacija med odjemalcem in strežnikom se začne tako, da odjemalec na vhodni tok strežnika pošlje 8 bajtov dolgo unikatno identifikacijsko številko razstavnega eksponata v obliki tabele bajtov. Strežnik jo prebere z zaporednimi klici metode *InputStream.read()*, ki z vhodnega toka prebere posamezen bajt.

```
Byte posamezni_byte = input.read();
```

Ko strežnik prebere identifikacijsko številko eksponata, z uporabo MySQL Java konektorja ustvari povezavo s podatkovno bazo in naredi poizvedbo, ki vrne opis in sliko razstavnega eksponata, ki se v podatkovni bazi nahaja pod prejeta identifikacijsko številko. Vrnjene podatke pretvorimo v tabelo bajtov in jih preko izhodnega toka pošljemo odjemalcu.

Priprava podatkov za pošiljanje:

Opis in sliko eksponata iz podatkovne baze dobimo s pomočjo rezreda *Statement* in *ResultSet*. Poizvedbo vpišemo kot parameter metode *Statement.executeQuery()*.

```
java.sql.Statement s = con_database.createStatement();
java.sql.ResultSet r = s.executeQuery("SELECT description,slika FROM
razstavni_eksponati WHERE Tag_id = '"+S_Tag_uid+"'");
```

Opis si zapomnimo v niz *S_tekst*, sliko pa si začasno shranimo, kot datoteko s končnico .jpg na lokalni disk računalnika.

```
File image1 = new File("C:\\Slike_java\\java.jpg"); //za hrambo slike
FileOutputStream fos = new FileOutputStream(image1);
```

```
while(r.next()) {
    String opis = r.getString(1);
    byte[] buffer = new byte[1];
    InputStream is = r.getBinaryStream(2);
```

V zanki beremo bajte in jih zapisujemo v datoteko s klicem metode *FileOutputStream.write()*:

```
while (is.read(buffer) > 0) {
    fos.write(buffer);
}
fos.close(); //zapremo FileOutputStream
}
```

Pridobljene podatke nato pretvorimo v tabelo bajtov, da jih lahko preko izhodnega toka pošljemo odjemalcu.

Zapis opisa razstavnega eksponata in njegove dolžine v tabelo bajtov:

```
byte [] BA_dolzina_teksta = S_dolzina_teksta.getBytes();
byte [] BA_Tekst = S_tekst.getBytes();
```

Branje slike iz diska:

```
BufferedImage image = ImageIO.read(new File("C:\\Slike_java\\java.jpg"));
ByteArrayOutputStream baos = new ByteArrayOutputStream();
ImageIO.write(image, "jpg", baos);
baos.flush();
```

Zapis slike in njene dolžine v tabelo bajtov:

```
byte [] BA_dolzina_slike = S_dolzina.getBytes();
byte [] BA_Slika = baos.toByteArray();
```

Podatke imamo sedaj zapisane v obliki tabele bajtov. Ker odjemalec ne ve koliko bajtov je namenjenih za opis in koliko za sliko, potrebujemo poleg opisa in slike, še dve spremenljivki. Spremenljivka *BA_dolzina_teksta*, odjemalcu pove koliko bajtov, ki jih prebere z vhodnega toka je namenjenih opisu, spremenljivka *BA_dolzina_slike* pa odjemalcu pove koliko bajtov je namenjenih zapisu slike.

Strežnik na izhodni tok najprej pošlje dolžino opisa v bajtih, sledi mu opis, nato dolžina slike in slika.

```
out.write(BA_dolzina_teksta,0,4); //prvi sterje bayti pomenijo dolzino!!!!
out.write(BA_Tekst,0,I_dolzina_teksta_v_bytih);
out.write(BA_dolzina_slike,0,4); //prvi sterje bayti pomenijo dolzino!!!!
out.write(BA_Slika,0,I_dolzina_slike_v_bytih);
```

5.2.2 MySQL Connector/J in uporaba MySql poizvedb

MySQL Connector/J je JDBC (ang. Java Database Connectivity) gonilnik, ki omogoča spojitev aplikacije razvite v programskem jeziku Java s podatkovno bazo MySQL. Dostopen je na spletnih straneh MySQL [19].

Uporaba Jave z MySQL podatkovno bazo:

1. Dodajanje gonilnika projektu v katerem razvijamo aplikacijo
2. Povezava Jave z bazo preko razreda `Driver.Manager`
3. Kreiranje SQL poizvedb v sklopu Java programa

Dodajanje gonilnika projektu v katerem razvijamo aplikacijo

Gonilnik si prenesemo s spletne strani in ga shranimo na lokalni disk. Nato pa v ustreznem projektu v okolju Eclipse nastavimo pot do gonilnika.

Povezava Jave z bazo preko razreda `Driver.Manager`

Če želimo v aplikaciji vzpostaviti povezavo s podatkovno bazo MySQL, moramo najprej uporabiti `MySQLConnector/J`, nato pa v programski kodi vzpostaviti povezavo. To naredimo z naslednjimi ukazi:

```
String driver = "com.mysql.jdbc.Driver";
String connection = "jdbc:mysql://localhost:3306/technical_museum";
String user = "root";
String password = "";
Class.forName(driver);
Connection con_database = null;
con_database = DriverManager.getConnection(connection, user, password);
```

Za vzpostavitev povezave s podatkovno bazo MySQL, moramo poklicati metodo `getConnection()` razreda `DriverManager`. Kot parametre potrebujemo URL naslov

podatkovne baze ter uporabniško ime in geslo uporabnika, ki ima pravico za uporabo podatkovne baze MySQL. S tem dobimo objekt razreda *Connection*, ki ga nato uporabljamo pri kreiranju SQL poizvedb.

Kreiranje SQL poizvedb v sklopu Java programa

Ko je povezava vzpostavljena, lahko začnemo pisati poizvedbe. Če želimo v bazo vstavljati podatke, ustvarimo primerek razreda *PreparedStatement*. Nato pokličemo metodo razreda *Connection.prepareStatement()* in ji kot parameter podamo ustrezno SQL poizvedbo.

Primer poizvedbe s stavkom INSERT :

```
String user_name = "Uporabnik (" + remoteAddress + ")";
String vstavi = "INSERT INTO statistics (tag_id,name,Bluetooth_id) VALUES (?, ?, ?)";

PreparedStatement pre = (PreparedStatement) con_database.prepareStatement(vstavi);
pre.setString(1, S_Tag_uid); //priprava podatkov
pre.setString(2, user_name);
pre.setString(3, remoteAddress);
pre.executeUpdate(); // izvede se update v tabeli statistics
pre.close(); // stavek je na koncu potrebno zapreti
```

Če iz podatkovne baze zahtevamo rezultat, pa to storimo s pomočjo instance razreda *Statement* in razreda *ResultSet*.

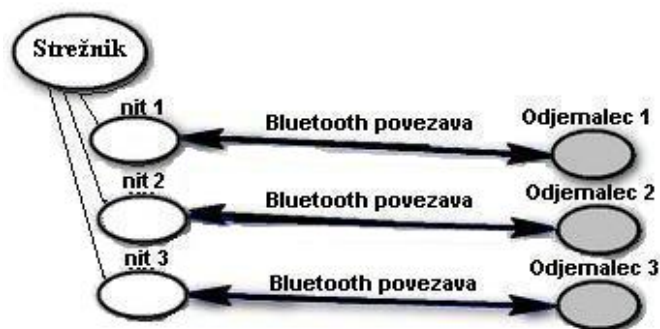
Primer poizvedbe s stavkom SELECT:

```
java.sql.Statement s = con_database.createStatement();
java.sql.ResultSet r = s.executeQuery ("SELECT name, description FROM
razstavni_eksponati ");

while(r.next()) { // v zanki beremo podatek za podatkom
    name = r.getString(1);
    opis = r.getString(2);
    /*sledi izpis ali zapis podatkov v tabelo */
}
```

5.2.3 Uporaba niti

Pri razvoju aplikacije za delo z več odjemalci, je potrebna uporaba niti (ang. Threads). Java aplikacija se začne izvajati v metodi *main()* in stavke izvajajo zaporedno. Niti uporabljamo zato, da določeni procesi lahko tečejo neodvisno od drugih. V aplikaciji so niti uporabljene za vzpostavitev povezave med Bluetooth odjemalcem in Bluetooth strežnikom (slika 22).



Slika 22. Komunikacija med Bluetooth odjemalci in strežnikom.

Osnovni koncept uporabe niti, je v Javi možen na dva načina. Razred lahko implementira vmesnik *Runnable* in nato metodo *run()*. Lahko pa podedujemo razred *Thread* in povozimo metodo *run()*. V metodo *run()* pa nato vnesemo obnašanje, ki ga želimo.

Aplikacija Bluetooth strežnika je sestavljena iz dveh razredov. Razred *J2SEServer*, poskrbi za uvodne nastavitve povezave Bluetooth in povezave s podatkovno bazo, nato pa v zanki čaka na odjemalce. Ko se odjemalec poveže, se ustvari primerek razreda *Server_komunikacija_s_clientom*, ki je izpeljan iz razreda *Thread()*. To naredimo s klicem *new Server_komunikacija_s_clientom (LocalDevice l, StreamConnection c, Connection c_database).start()*; Kot parametre, razred potrebuje objekt razreda *LocalDevice*, povezavo s podatkovno bazo in povezavo z oddaljeno Bluetooth napravo. Po kreiranju razreda, strežnik nadaljuje s čakanjem odjemalcev, omenjeni razred pa vodi potek komunikacije med povezanim odjemalcem, strežnikom in podatkovno bazo.

5.3 Razvoj aplikacije za mobilni telefon

Za razvoj mobilne aplikacije v Javi ME je bilo potrebno namestiti razvojno okolje Eclipse in dodatek Eclipse ME. Poleg tega pa so bila uporabljena še orodja J2ME Wireless Toolkit in Nokia 6212 Series 40 NFC SDK, ki se ju integrira v okolje Eclipse.

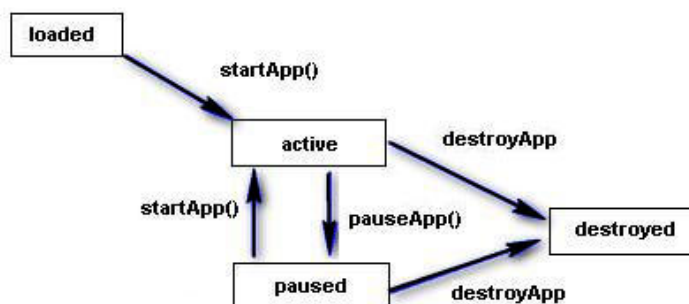
Potrebno je bilo implementirati mobilno aplikacijo oziroma MIDlet za mobilni telefon NOKIA 6212 Classic NFC. Naloga aplikacije je, da iz RFID značke, ki se nahaja na razstavnem eksponatu prebere unikatno identifikacijsko številko in jo preko Bluetooth komunikacije posreduje strežniku. Ko ji strežnik odgovori z imenom, sliko in opisom razstavnega eksponata, pa ga prikaže na zaslonu mobilne naprave.

MIDlet aplikacija mora izhajati iz osnovnega razreda *javax.microedition.midlet.MIDlet*, ki vsebuje ključne metode, skozi katere se nadzira življenjski cikel MIDlet aplikacije, ki je prikazan na sliki 23.

Metode, ki jih mora vsebovati vsak MIDlet so :

- *startApp()* // se pokliče ob zagonu aplikacije
- *pauseApp()* // omogoča začasno prekinitev aplikacije, sprostijo se zasedeni viri
- *destroyApp()* // omogoča dokončno ustavitev aplikacije

MIDlet se lahko nahaja v štirih različnih stanjih: stanje nalaganja (ang. loaded), aktivno (ang. active), ustavljeno (ang. paused) in uničeno (ang. destroyed) stanje. Ko se aplikacija nalaga na sistem, je v stanju nalaganja. To je čas do tedaj, ko upravljalca programov ne pokliče metode *startApp()*. Takrat MIDlet preide v aktivno stanje. V aktivnem stanju se nahaja do tedaj, ko se pokliče metoda *pauseApp()*, ki pomeni sprostitvev zasedenih virov ali metoda *DestroyApp()*, ki delovanje aplikacije dokončno zaključi.



Slika 23. Življenski cikel MIDlet aplikacije.

Za branje in delo z RFID značkami, je v Javi ME na voljo vmesnik za brezkontaktno komunikacijo, ki ga uporabimo pri razvoju aplikacije.

5.3.1 Vmesnik za brezkontaktno komunikacijo (JSR 257)

Je programski vmesnik (API) namenjen razvoju MIDlet aplikacij, ki vključujejo delo z brezkontaktno komunikacijo oziroma s tehnologijo NFC. Vmesnik podpira branje in pisanje podatkov shranjenih na enotah in napravah, ki so prilagojene za brezkontaktno komunikacijo. Sestavljen je iz paketov, ki so opisani v tabeli 5.

Paket	Opis
<i>javax.microedition.contactless</i>	omogoča osnovni nabor funkcionalnosti
<i>javax.microedition.contactless.ndef</i>	omogoča prenos podatkov, ki so narejene po formatu NFC Data Exchange Format (NDEF), med različnimi napravami
<i>javax.microedition.contactless.rf</i>	vmesnik za fizične radiofrekvenčne enote
<i>javax.microedition.contactless.sc</i>	omogoča komunikacijo s pametnimi karticami
<i>javax.microedition.contactless.visual</i>	omogoča branje informacij z vizualnih enot (črtnih kod) in njihovo generiranje

Tabela 5. Paketi, ki jih vsebuje vmesnik JSR 257.

V mobilni aplikaciji, bo vmesnik uporabljen za branje unikatne identifikacijske številke (UID) z RFID značk. Branje poteka v dveh korakih:

1. Odkrivanje brezkontaktne značke,
2. Branje podatkov z brezkontaktne značke.

Odkrivanje brezkontaktne značke

Za odkrivanje značk je v vmesniku na voljo razred `javax.microedition.contactless`, ki vsebuje razrede in vmesnike, ki so opisani v tabeli 6.

Razred/Vmesnik	Opis
<i>Tag Connection</i>	vmesnik za povezavo vseh RFID značk in pametnih kartic
<i>TargetListener</i>	vmesnik, ki zagotavlja mehanizem za zaznavanje brezkontaktnih ciljev
<i>TransactionListener</i>	vmesnik, ki vsebuje varnostne elemente
<i>DiscoveryManager</i>	razred, ki nudi mehanizme za zaznavanje brezkontaktnih ciljev
<i>TargetProperties</i>	razred, ki zbira skupne lastnosti podprtih ciljev.
<i>TargetType</i>	razred, ki zbira brezkontaktne cilje podprte s strani vmesnika jsr 257
<i>ContactlessException</i>	izjema, ki je sprožena, ko se skuša izvesti nepodprta operacija.

Tabela 6. Tabela razredov in vmesnikov, ki so vsebovani v paketu `javax.microedition.contactless`.

Sporočilo, ki se bo prenašalo iz RFID značke na mobilno napravo ima obliko NFC Data Exchange Format. To je osnovni format sporočil, ki se prenašajo med NFC napravami in je bil določen v okviru NFC Foruma.

```
try {
    DiscoveryManager dm = DiscoveryManager.getInstance();
    dm.addTargetListener (this, TargetType.NDEF_TAG);
}
catch (ContactlessException ce) {
    //rokovna je z izjemo
}
```

Z zgornjimi vrsticami kode odkrivamo značke, ki so podprte s strani NFC Foruma. Najprej ustvarimo objekt razreda *DiscoveryManager* in mu dodamo poslušalca *TargetListener*. Kot parameter poslušalca podamo tip iskanega cilja, ki je v našem primeru *TargetType.NDEF_TAG*.

Ko naprava zazna ustrezno značko, JVM spoži metodo *targetDetected()*. Metoda ima kot parameter tabelo *targetDetected[]*, v kateri so hranjeni podatki o značkah, ki jih je naprava zaznala.

```
public void targetDetected(TargetProperties[] targetProperties) {
    if (targetProperties.length == 0) {
        return; // v primeru da ni nobene značke, se metoda zaključi
    }
}
```

Branje podatkov z brezkontaktne značke

V metodi *targetDetected()*, nato rokujemo s podatki, ki so shranjeni na RFID značkah. V našem primeru je potrebno z značke prebrati UID. To storimo tako, da iz tabele *TargetProperties*, ki hrani podatke o prebranih značkah, pokličemo metodo *getUID()*.

```
TargetProperties tmp = targetProperties[0];
String S_Tag_uid = tmp.getUid();
```

UID številka, se nato preko Bluetooth povezave pošlje Bluetooth strežniku.

5.3.2 Bluetooth odjemalec

Pri implementaciji Bluetooth odjemalca, je bil uporabljen vmesnik JSR 82, ki je podprt s strani mobilnega telefona Nokia 6212 Classic.

Osnovni koncept Bluetooth aplikacije, ki ima vlogo odjemalca je sestavljen iz naslednjih komponent:

1. Inicializacija iskalca servisov (ang Discovery Agent), ki omogoča pridobitev zapisa servisov (ang. service record),
2. Konstruiranje URL naslova, ki se izvede s pomočjo zapisa servisov,
3. Vzpostavitev povezave s strežnikom,
4. Sprejemanje in pošiljanje podatkov med odjemalcem in strežnikom.

Inicializacija iskalca servisov, ki omogoča pridobitev zapisa servisov

Ker naša Bluetooth naprava ne ve, kaj so ostale naprave v območju in katere servise nudijo, mora to ugotoviti. Za iskanje Bluetooth naprav, sta v vmesniku na voljo razred *DiscoveryAgent* in vmesnik *Discovery Listener*. Objekt razreda *DiscoveryAgent* pridobimo tako, da na lokalni napravi pokličemo metodo *LocalDevice.getDiscoveryAgent()*. Z objektom razreda *DiscoveryAgent* pridobimo seznam naprav dostopnih v okolici.

```
LocalDevice localdevice = LocalDevice.getLocalDevice();
DiscoveryAgent discoveryAgent = localdevice.getDiscoveryAgent();
```

Če želimo poiskati naprave v okolici, mora naš razred implementirati vmesnik *Discovery Listener*. Razred, ki implementira ta vmesnik mora vsebovati 4 metode:

- *deviceDiscovered(remoteDevice arg0, DeviceClass arg1)* je metoda, ki se jo pokliče takrat, ko lokalna Bluetooth naprava najde novo oddaljeno Bluetooth napravo.
- *inquiryCompleted(int discType)* je metoda, ki se pokliče, ko je iskanje oddaljenih Bluetooth naprav zaključeno. Polje *discType* lahko zavzame tri vrednosti.
 - *DiscoveryListener.INQUIRY_COMPLETED*
 - *DiscoveryListener.INQUIRY_TERMINATED*
 - *DiscoveryListener.INQUIRY_ERROR*

- *servicesDiscovered(int arg0, ServiceRecord[] servRecord)* je metoda, ki se pokliče, ko lokalna naprava najde servis na oddaljeni napravi.
- *serviceSearchCompleted(int transID, int respCode)* je metoda, ki se pokliče, ko je iskanje servisov zaključeno. Polje *respCode* lahko zavzame pet vrednosti:
 - *DiscoveryListener.SERVICE_SEARCH_TERMINATED*
 - *DiscoveryListener.SERVICE_SEARCH_ERROR*
 - *DiscoveryListener.SERVICE_SEARCH_NO_RECORDS*
 - *DiscoveryListener.SERVICE_SEARCH_DEVICE_NOT_REACHABLE*
 - *DiscoveryListener.SERVICE_SEARCH_COMPLETED*

Vmesnik *Discovery Listener* deluje enako, kot drugi poslušalci. Preiskuje območje in ko napravo najde, sproži dogodek. Na začetku iskanja je potrebno poklicati metodo *DiscoveryAgent.startInquiry()*, ki lokalni napravi omogoči iskanje oddaljenih naprav. Ko napravo najde pokliče metodo *deviceDiscovered()*, ki pridobi objekt razreda *RemoteDevice*. Iskanje naprav poteka tako dolgo, dokler agent v območju ne najde nobene naprave več. Takrat pokliče metodo *inquiryCompleted()* in iskanje naprav se s tem zaključí. S tem smo pridobili seznam naprav v okolici. Sedaj je potrebno ugotoviti še servise, ki jih oddaljene naprave nudijo. To naredimo s klicem metode razreda *DiscoveryAgent.searchServices()*. Parametri te metode so opisani v tabeli 7.

```
public int searchServices(int[] attrSet, UUID[] uuidSet,
    RemoteDevice btDev, DiscoveryListener discListener)
    throws BluetoothStateException
```

Parameter	Opis
<i>int[] attrSet</i>	ko najdemo ustrezni servis, nam atributi specificirani v tabeli <i>attrSet</i> podajo osnovne lastnosti odkritih lastnosti
<i>UUID[] uuidSet</i>	parameter, ki določi katere UUID iščemo
<i>RemoteDevice btDev</i>	oddaljena Bluetooth naprava, na kateri iščemo servise
<i>DiscoveryListener discListener</i>	poslušalec, ki proži metodo ob odkritju ustreznega servisa

Tabela 7. Parametri metode *searchServices*.

Iskanje servisov poteka podobno kot iskanje naprav. Ko najdemo servis, se pokliče metoda *servicesDiscovered()*, ki vrne zapis posameznega servisa. Zapis mora ustrezati določenim pravilom, ki jih v kodi definiramo. Na primer v našem primeru bi iskali servis, ki ima ime servisa in UUID enak kot UUID številka in ime na strežniku, poleg tega pa mora za povezovanje uporabljati protokol RFCOMM. Iskanje servisov poteka vse dokler agent na napravah ne najde nobenega ustreznega servisa več. Takrat pokliče metodo *serviceSearchCompleted()*.

Konstruiranje URL naslova, s pomočjo zapisa servisov

Ko je celotna preiskava končana, imamo na razpolago tabelo zapisov servisov, ki ustrezajo navedenim pogojem. Potrebno je samo še izbrati servis in poklicati metodo

ServiceRecord.getConnectionURL(). S klicem te metode pridobimo povezovalni URL naslov, ki služi vzpostavljanju povezave.

```
String url = servRecord[i].getConnectionURL(int security, boolean master);
```

Iskanje naprav in servisov na oddaljenih napravah se v psevdo kodi opiše nekako tako:

1. Kreiranje agenta za raziskovanje (ang. *DiscoveryAgent*),
2. Začetek iskanja oddaljenih Bluetooth naprav,
3. Shranjevanje vsake odkrite oddaljene naprave v vektor,
4. Čakanje, da agent zaključi preiskavo,
5. Za vsako odkrito napravo agent poišče vse servise in jih shranjuje v tabelo zapisov servisov (ang. *serviceRecord*),
6. Vrne tabelo najdenih zapisov servisov.

Ta pot iskanja je edina v primeru, ko ne vemo katero napravo in servis iščemo. Potrebno je uporabiti celoten proces iskanja, čeprav je zamuden. V primeru, ko pa nam je vseeno katera naprava nam ponudi določen servis, lahko uporabimo metodo *selectService()*. To je metoda razreda *DiscoveryAgent*, ki poišče določen servis na napravi, ki za nas ni pomembna. Kot parametre moramo metodi dodati univerzalno unikatno identifikacijsko številko servisa, ki se mora ujemati s številko servisa na strežniku.

```
String url = DiscoveryAgent.selectService(UUID uuid, int security, boolean master);
```

Vrednost, ki nam jo omenjena metoda vrne (v primeru da najde servis), je prav tako kot pri metodi *ServiceRecord.getConnectionURL()* povezovalni URL naslov, ki se ga nato uporabi pri povezovanju z Bluetooth strežnikom.

Vzpostavitev povezave s strežnikom

Povezavo odpremo s klicem metode *Connector.open()*, ki mu kot parameter podamo povezovalni URL naslov.

```
StreamConnection conn = (StreamConnection) Connector.open(url);
```

Ko odjemalec vzpostavi povezavo, dobimo prav tako kot pri strežniku objekt razreda *StreamConnection*. S tem objektom nato upravljamo tokove med lokalno in oddaljeno Bluetooth napravo.

Sprejemanje in pošiljanje podatkov med odjemalcem in strežnikom

Tudi tukaj ravno tako, kot pri strežniku s klicem spodnjih dveh metod kreiramo vhodni in izhodni tok.

```
InputStream input = conn.openInputStream();  
OutputStream output = conn.openOutputStream();
```

S klicem metode *openOutputStream()* ustvarimo izhodni tok, s katerim pošiljamo sporočila na strežnikovo Bluetooth napravo. S klicem metode *openInputStream()* pa ustvarimo vhodni tok, s katerim sprejemamo sporočila iz Bluetooth naprave strežnika.

Strežnik odjemalcu kot prvi podatek pošlje dolžino opisa v bajtih, ta jo prebere in na njeni podlagi v zanki izvaja branje in shranjevanje opisa v tabelo bajtov *BA_tekst*.

```
int f=0;
while (f!= BA_tekst.length) {
    int ch = in.read(BA_tekst, f, BA_tekst.length - f);
    f += ch;
}
```

Po prebranem opisu, pa enako naredimo še s sliko. Sliko si zapomnimo v tabelo bajtov *BA_slika*.

```
Int k = 0;
while (k != BA_slika.length) {
    int ch = in.read(BA_slika, k, BA_slika.length - k);
    k += ch;
}
```

Za prikaz na zaslonu, je tabeli bajtov potrebno spremeniti v ustrezna podatkovna tipa. Za sliko je to podatkovni tip *Image*, za opis pa podatkovni tip niz (ang. *String*).

```
Image image = Image.createImage(BA_slika,0,I_dolzina_slike2); //kreiranje slike
String tekst = new String(BA_tekst); //kreiranje niza iz tabele bajtov
```

Prikaz slike in opisa na zaslonu mobilne naprave:

Za prikaz podatkov uporabimo funkcijo *Form.append()*.

```
f.append(new ImageItem(null,image,ImageItem.LAYOUT_NEWLINE_BEFORE |
    ImageItem.LAYOUT_CENTER |
    ImageItem.LAYOUT_NEWLINE_AFTER, null));

String tekst = new String(BA_tekst);
f.append(tekst);
```


6. Predstavitev aplikacij in namestitev

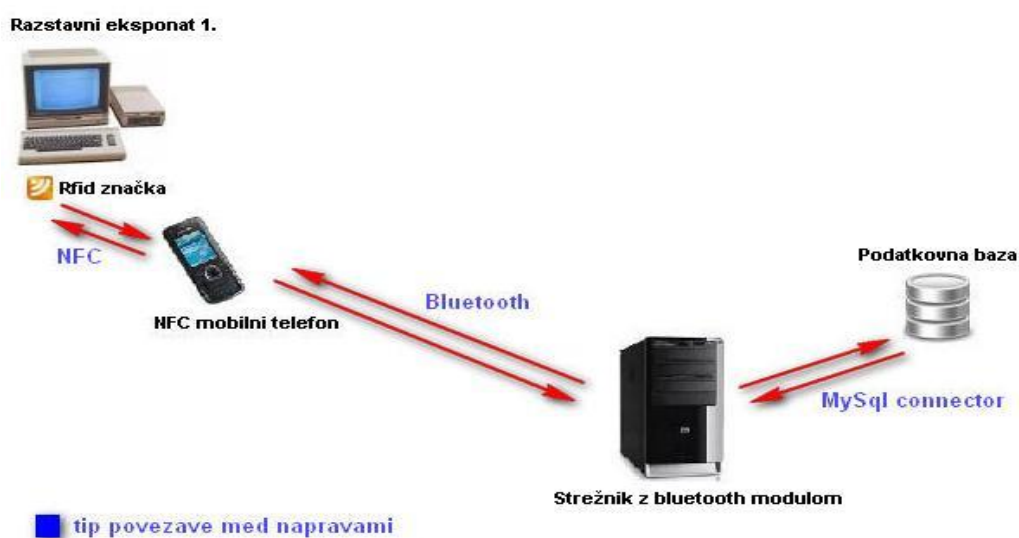
6.1 Aplikacije za interaktiven ogled razstav

Aplikacija za interaktiven ogled razstav, je namenjena obiskovalcem in vodstvu muzeja oziroma galerije. Obiskovalci bi z uporabo aplikacije lahko pridobili podrobnejše informacije o muzeju, oddelkih in razstavnih eksponatih in bili s tem deležni celovitejše uporabniške izkušnje, kot so jih bili ob obiskih muzejev vajeni do sedaj. Vodstvo bi dobilo podatke o aktivnostih obiskovalcev, saj se vsak dogodek, ki ga s pomočjo aplikacije sproži obiskovalec zabeleži v podatkovno bazo. Tako bi vodstvo dobilo podatke o najpogostejše obiskanih oddelkih in razstavnih eksponatih. Na podlagi teh podatkov, bi lahko dopolnili muzej z oddelki in razstavnimi eksponati v smeri, ki jo obiskovalci želijo, nezanimive oddelke in eksponate pa bi iz muzeja odstranili.

Za namestitev aplikacije v realnem okolju, bi potrebovali ustrezno brezžično omrežje, delujoč strežnik in na njem nameščen del aplikacije, ki deluje kot strežnik med mobilnimi napravami in podatkovno bazo. Poleg tega pa bi vsak obiskovalec potreboval še ustrezno mobilno napravo. Tu imamo dve možnosti:

- V muzeju ali galeriji bi imeli določene mobilne naprave, z nameščeno aplikacijo, ki bi jih izposojali obiskovalcem.
- Obiskovalec ima mobilno napravo z NFC čipom. S spleta prenese aplikacijo in jo naloži.

Če želimo, da bo aplikacija delovala, moramo najprej postaviti ustrezno Bluetooth omrežje in na osebni računalnik pognati sistema za upravljanje z relacijskimi podatkovnimi bazami MySQL. Poleg tega pa moramo pognati še aplikacijo, ki na osebni računalnik deluje kot Bluetooth strežnik. Ko to aplikacijo poženemo, se ustvari zapisu Bluetooth servisa, na katerega se odjemalci lahko začnejo povezovati. Potek komunikacije med odjemalcem in strežnikom je prikazan na sliki 24.



Slika 24. Načini komunikacije med napravami.

V aplikaciji se podatki prenašajo med odjemalcem, strežnikom in podatkovno bazo. Odjemalec (NFC mobilni telefon) mora kot prvo vzpostaviti Bluetooth povezavo s strežnikom (Osební računalnik). Po uspešno vzpostavljeni povezavi, komunikacija poteka tako, da odjemalec čaka na uporabnika, da se z NFC mobilno napravo približa eni izmed RFID značk, ki se nahajajo v muzeju. Ko se to zgodi odjemalec iz RFID značke prebere UID številko in jo preko Bluetooth komunikacije posreduje strežniku. Strežnik UID sprejme, naredi zapis v tabelo 'statistika' v podatkovni bazi in nato na podlagi UID številke naredi še poizvedbo v tabeli 'razstavni eksponati'. Podatke, ki jih vrne poizvedba SQL, v našem primeru sta to slika in opis razstavnega eksponata, nato preko Bluetooth posreduje odjemalcu. Odjemalec podatke sprejme in jih prikaže na zaslonu mobilne naprave. Komunikacija se ponavlja dokler obiskovalec ne zaključi obiska ali ugasne aplikacije.

6.2 Aplikacija za administracijo podatkovne baze in pregled statistike

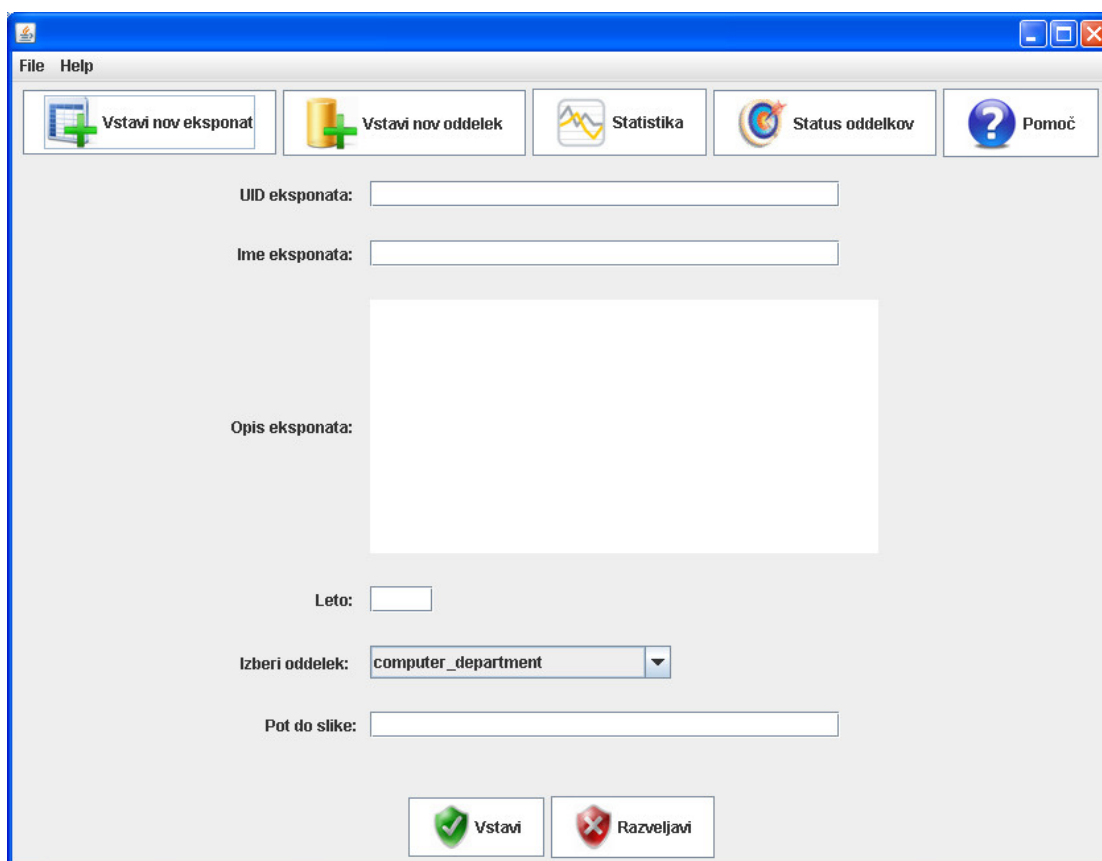
Aplikacija za administracijo podatkovne baze in spremljanje statistike je napisana v jeziku Java SE. Grafični vmesnik je sestavljen iz menija, treh obrazcev in forme za pregled statistike. S pomočjo menija, aplikacijo lahko zaključimo ali si ogledamo osnovne podatke o aplikaciji. Obrazci in forma pa so uporabniku na voljo za:

- Vpis novih oddelkov oziroma razstav,
- Vpis novih razstavnih eksponatov,
- Ogled in spreminja status posameznih oddelkov,
- Ogled statistike.

Za vpis novega razstavnega eksponata kliknemo na tipko vstavi nov eksponat. Prikaže se nam obrazec (slika 25), ki je sestavljen iz:

- Polja za vpis unikatne identifikacijske številke razstavnega eksponata,
- Polja za vpis imena razstavnega eksponata,
- Polja za vpis opisa razstavnega eksponata,
- Polja za vpis leta v katerem je bil razstavni eksponat narejen,
- Polje za vpis poti, kjer se nahaja slika razstavnega eksponata.

Poleg teh polj, obrazec vsebuje še izbirni meni, v katerem je potrebno izbrati oddelek, v katerega želimo vpisati nov razstavni eksponat. Ko smo izpolnili vsa polja, moramo samo še klikniti na tipko Vstavi v spodnjem delu obrazca. Če smo vse podatke vnesli pravilno in je povezava s podatkovno bazo vzpostavljena, se izpiše obvestilo, da se je razstavni eksponat pravilno vnesel v bazo. V nasprotnem primeru, pa obvestilo o spodletelem poizkusu. Tipka Razveljavi je namenjena hitrejšemu brisanju vseh polj, če se je uporabnik zmotil pri vpisovanju vrednosti.



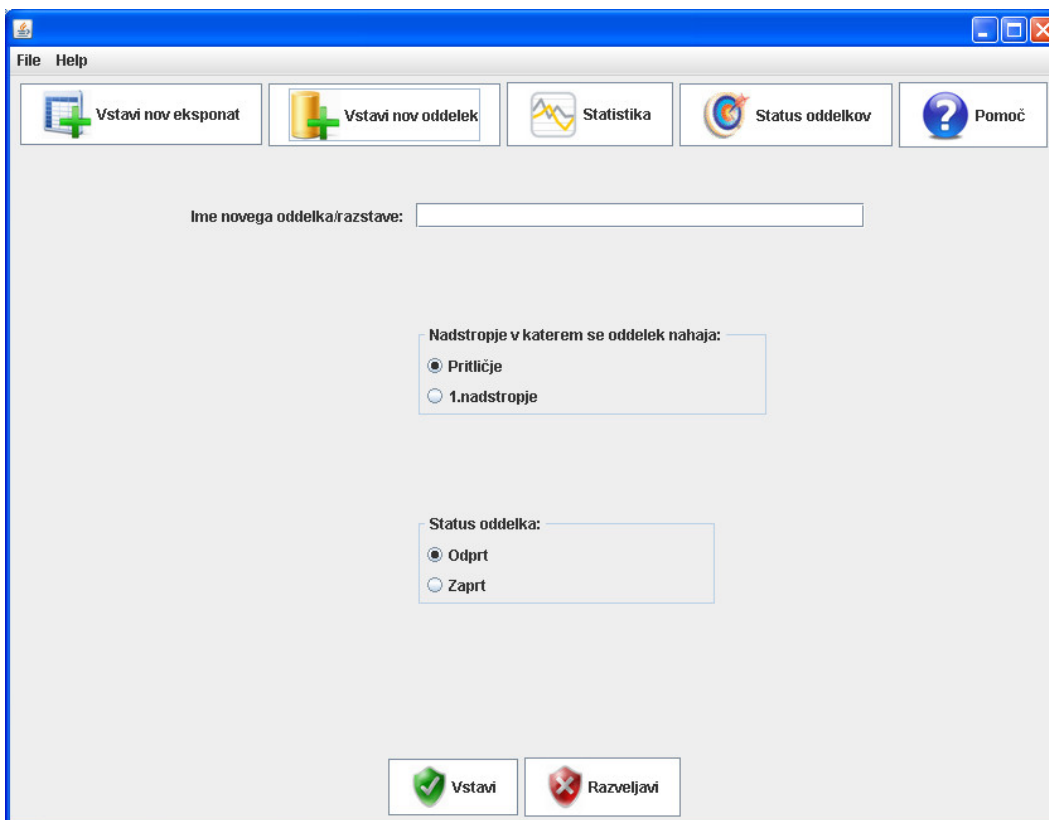
Slika 25. Obrazec za vnos novih razstavnih eksponatov.

Za vpis novega oddelka oziroma razstave, kliknemo na tipko vstavi nov oddelek. Prikaže se nam obrazec (slika 26), ki je sestavljen iz:

- Polja za vpis imena novega oddelka oziroma razstave,
- Izbirnih tipk za izbor nadstropja v katerem se nov oddelke ali razstava nahaja,
- Izbirnih tipk za izbor statusa oddelka.

Obrazec se izpolni tako, da uporabnik najprej vpiše ime oddelka, nato izbere nadstropje v katerem se oddelke nahaja, izbere še status oddelka in klikne na tipko vstavi. Če so podatki vpisani in je povezava s podatkovno bazo vzpostavljena, se bodo vnesli v podatkovno bazo. Izpisalo se bo tudi besedilo, ki poda sporočilo ali je bil oddelek uspešno dodan v podatkovno bazo.

Za nastavitve statusa posameznih oddelkov kliknemo na tipko Status oddelkov. Prikaže se obrazec (slika 27), ki ga sestavljajo po dve izbirni tipki (odprt, zaprt) za vsak oddelek. Če želimo spreminjati status posameznega oddelka, izberemo ustrezen status in kliknemo na tipko Spremeni v spodnjem delu obrazca. S tem lahko dosedanji status oddelkov, zamenjamo z novo izbranim statusom.



File Help

Vstavi nov eksponat Vstavi nov oddelek Statistika Status oddelkov Pomoč

Ime novega oddelka/razstave:

Nadstropje v katerem se oddelek nahaja:

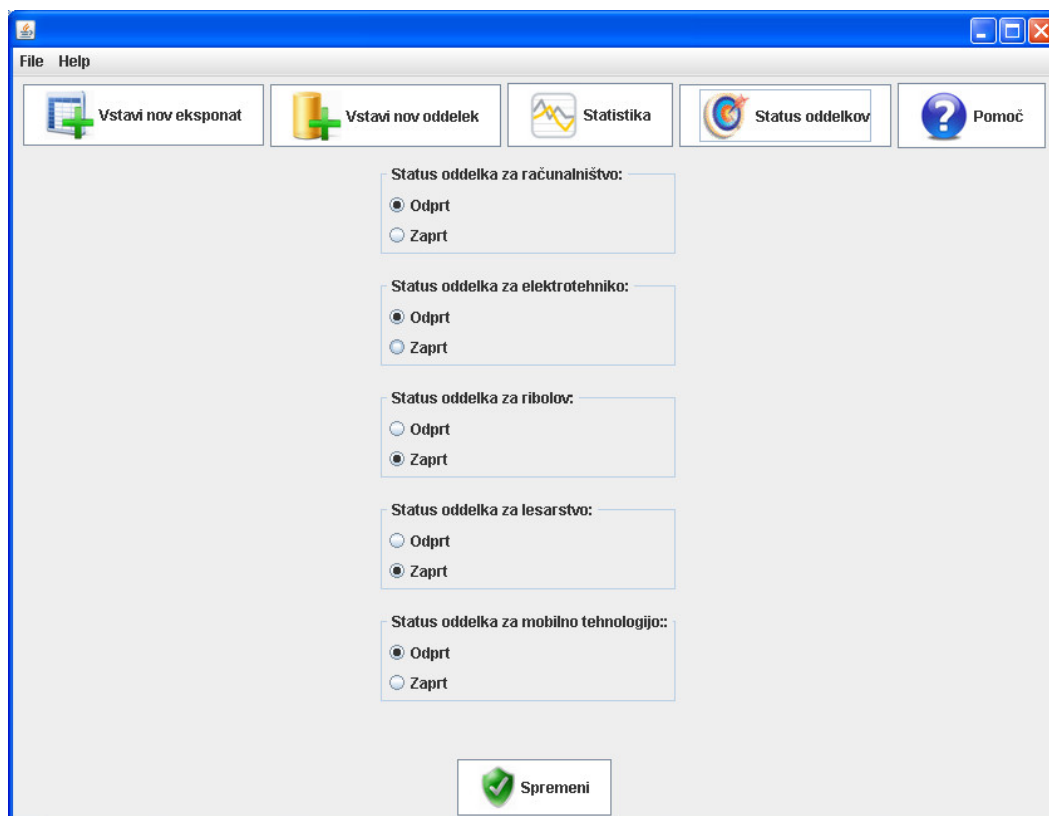
Prilličje
 1.nadstropje

Status oddelka:

Odprt
 Zaprto

Vstavi Razveljavi

Slika 26. Obrazec za vnos novih oddelkov oziroma razstav.



File Help

Vstavi nov eksponat Vstavi nov oddelek Statistika Status oddelkov Pomoč

Status oddelka za računalništvo:

Odprt
 Zaprto

Status oddelka za elektrotehniko:

Odprt
 Zaprto

Status oddelka za ribolov:

Odprt
 Zaprto

Status oddelka za lesarstvo:

Odprt
 Zaprto

Status oddelka za mobilno tehnologijo:

Odprt
 Zaprto

Spremeni

Slika 27. Obrazec za nastavitve statusa posameznim oddelkom.

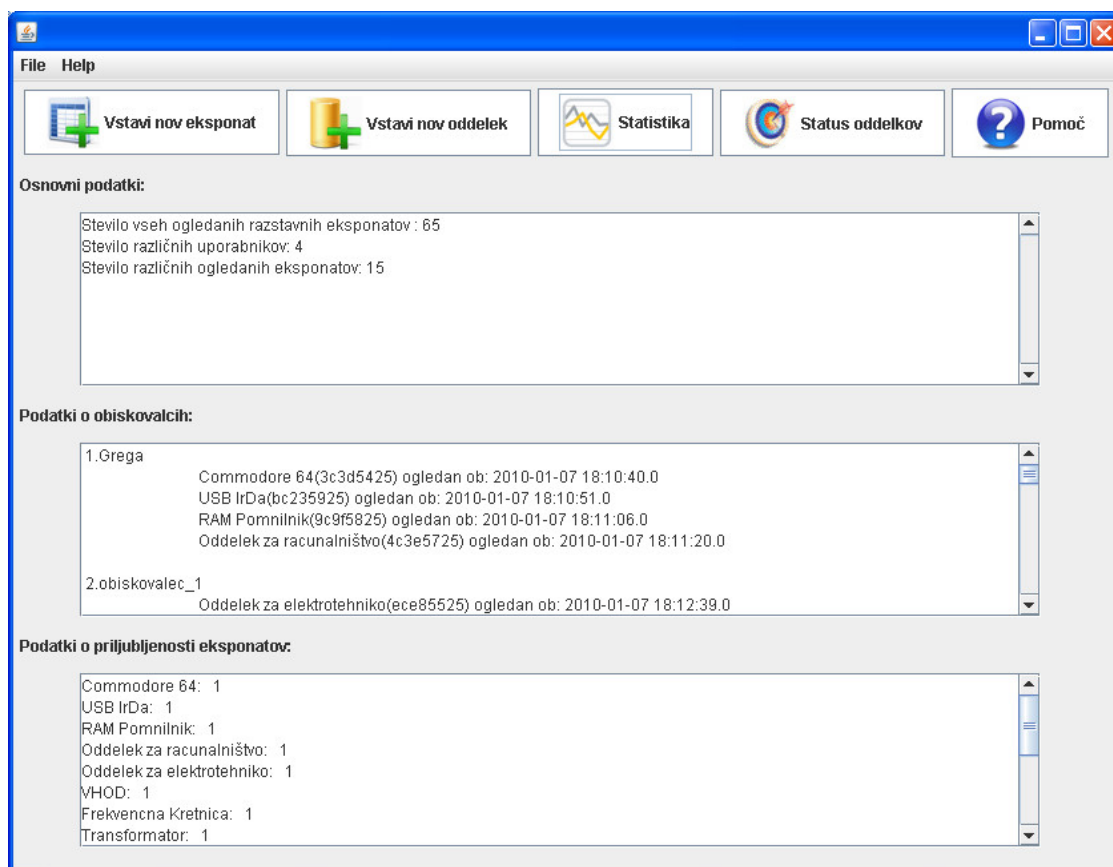
Za ogled statističnih podatkov o muzeju, kliknemo na tipko Statistika. Prikaže se forma, ki je sestavljena iz treh področij (slika 28):

- Osnovni statistični podatki,
- Podatki o obiskovalcih,
- Podatki o priljubljenosti razstavnih eksponatov.

Prvo področje prikazuje koliko je vseh ogledanih razstavnih eksponatov, koliko obiskovalcev si je eksponate ogledovalo in koliko različnih eksponatov je bilo ogledanih.

Drugo področje prikazuje podatke o obiskovalcih. Prikaže imena vseh obiskovalcev in eksponate, ki so si jih ogledali. Dodan je tudi čas ogleda vsakega eksponata.

Tretje področje prikazuje koliko ogledov je imel določen razstavni eksponat. Na podlagi tega vidimo, kateri eksponati so v muzeju najbolj priljubljeni.



Slika 28. Forma za pregled statistike.

6.3 Spletna stran

Na sliki 29 je prikazan potek komunikacije med podatkovno bazo, strežnikom in spletnim odjemalcem.



Slika 29. Komunikacije med odjemalcem, strežnikom in podatkovno bazo.

Ko odjemalec v internetni brskalnik vpiše naslov spletne strani, (v našem primeru je to http://localhost/Portal_museum/my_page.html) spletni strežnik Apache zazna zahtevek. Na podlagi zahtevka, zgenerira spletno stran v XHTML in jo posreduje odjemalcu. Stran se zgenerira tako, da ustrezno interpretira PHP kodo, ki vsebuje poizvedbe iz podatkovne baze tehničnega muzeja in iz nje naredi kodo XHTML.

Osnovni namen spletne strani je predstavitev tehničnega muzeja NFC-TM. Izdelana je v programskem jeziku XHTML (ang. Extensible HyperText Markup Language) v povezavi s PHP (ang. Hypertext Preprocessor ali Personal home page tools) in CSS (ang. Cascading Style Sheets). Sestavljena je iz osnovne strani in podstrani, v katerih so po oddelkih predstavljeni razstavniki eksponati. Na osnovni strani, ki je prikazana na sliki 30, so opisani osnovni podatki o muzeju NFC-TM in zemljevid. Na njej so prisotne tudi povezave do posameznih oddelkov:

- Oddelek za računalništvo,
- Oddelek za elektrotehniko,
- Oddelek za Mobilno telefonijo,
- Oddelek za lesarstvo,
- Oddelek za Ribolov.



Made 4 Janur 2009
 by Gregor Mlakar.

Slika 30. Spletna stran tehničnega muzeja NFC-TM.

S klikom na eno izmed povezav za oddelke, se ustvari nova stran, ki vsebuje imena, opise in slike razstavnih eksponatov, ki jih lahko najdemo na ustreznem oddelku v muzeju. Stran se ustvari tako, da s pomočjo PHP kode ustvari povezavo s podatkovno bazo tehničnega muzeja, iz nje prebere imena in opis razstavnih eksponatov in jih prikaže na spletni strani. Na sliki 31 je prikazan oddelček za mobilno telefonijo z opisi eksponatov.

- Home
- Oddelek za Računalništvo
- Oddelek za Elektrotehniko
- Oddelek za Mobilno telefonijo
- Oddelek za Lesarstvo
- Oddelek za Ribolov

Oddelek za mobilno telefonijo

Eksponati, ki se nahajajo na oddelku so:

Panasonic G50

Description from NFC: Panasonic G50 is a tri-band candy bar styled mobile phone, which is not only suitable for global roaming but is also one of the smallest and lightest models around, and so carrying it about will not weigh you down. Useful features of this model include GPRS and WAP technology, speakerphone for hands-free conversations, EMS messaging, and several polyphonic ringtones. The display in the G50 is a LCD offering a resolution of 128 x 96 pixels, and it can also reproduce up to 4,096 different colours, which gives clear viewing of the menus, pictures and games that you can use on this model. As this phone is capable of sending and receiving EMS messages, it is possible to use it to communicate messages containing a combination of simple images, sounds and text.



Nokia 8210

Description_from NFC_tag..The Nokia 8210 was, at the time of its release in 1999, the smallest, lightest Nokia mobile phone on the market,[citation needed] thus its selling point was based on its design and customization, with removable Xpress-on covers. Six differently coloured Xpress-on covers are available, as well as many third party ones. Unfortunately one of the design faults of the Nokia "82XX" series was that many developed "screen fade" which is a condition where figures on the display would eventually be so dim that it was impossible to read the display.



Slika 31. Stran, ki prikazuje oddelek za mobilno telefonijo.

XHTML je označevalni jezik za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. Od HTML se razlikuje predvsem po strožjih pravilih pisanja in usklajenosti s sintakso XML (ang. Extensible Markup Language). Dokument napisan v XHTML, prav tako kot HTML pišemo z značkami. Z njimi definiramo odseke strani, dele besedila, vstavljamo slike, animacije, tabele in ustvarjamo obrazce. Značke večinoma nastopajo v parih, potrebno jih je pisati med odpiralnima (Primer: <p>) in zapiralnima (Primer: </p>) znakoma. Osnovna stran napisana v jeziku XHTML zgleda tako:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Naslov</title>
<link rel="stylesheet" href="mystyle.css" //stil se nahaja v datoteki mystyle.css
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250"/>
</head>
<body>
<p>Prvi odstavek</p>
</body>
</html>

```

Z DOCTYPE določimo vrsto dokumenta, ki določa pravila in način prikaza. Sledi glava `<head>`, katera vsebuje naslov dokumenta `<title>`, pot do CSS datoteke in meta `<meta>` podatke, v katerih določimo nabor znakov na strani. Glavi sledi telo `<body>` dokumenta, v katerem je opisana dejanska vsebina strani in zaključek značk `</body>` in `</html>`.

S CSS določamo predloge oziroma izgled spletne strani. Z njim določamo pisavo, velikost črk in vizualno predstavitev spletne strani. XHTML predstavlja osnovno strukturo in hierarhijo spletne strani, medtem ko CSS predstavlja vizualno podobo.

PHP (ang. Hypertext Preprocessor ali Personal home page tools) je razširjen skriptni, objektno usmerjen, odprtokodni programski jezik, ki se uporablja za prikaz dinamičnih spletnih vsebin. PHP koda se ne prevaja v izvedljivo obliko, interpretira se na strežniku in generirano spletno stran pošlje odjemalcu. PHP datoteke imajo končnico `php`, v njih pa se poleg PHP kode, nahaja tudi XHTML koda. PHP je bil pri izdelavi spletne strani uporabljen za povezavo s podatkovno bazo in izpis podatkov o razstavnih eksponatih. Povezavo s podatkovno bazo vzpostavimo s funkcijama `mysql_connect()` in `mysql_select_db()`. Prva je namenjena povezovanju in potrebuje vhodne parametre naslov, uporabniško ime in geslo uporabnika, ki ima pravico do uporabe podatkovne baze. S funkcijo `mysql_select_db()`, pa izberemo ustrezno podatkovno bazo.

Začetek PHP kode se začne z blokom `<?PHP`, konča pa z znakoma `?>`. Vse, kar se nahaja med tema dvema blokoma, bo tolmač interpretiral kot PHP kodo. Spodnja koda prikazuje primer povezave s podatkovno bazo, SQL poizvedbo in izpis vrstic s pomočjo jezika PHP.

```
<?php
mysql_connect("localhost","root","");
mysql_select_db("technical_museum");

$result = mysql_query("SELECT * FROM razstavni_eksponati WHERE name =
'Oddelek za mobilno telefonijo' ");

while($row = mysql_fetch_array($result)){
echo "<p><h1>",$row['name'],"</h1></p>";
}
?>
```

6.4 Namestitev aplikacij v realnem okolju

Strežniška aplikacija se namesti na lokalni osebni računalnik ali strežnik, ki nudi vso potrebno podporo, za njeno izvajanje:

- Bluetooth omrežje ali adapter,
- Java SE,
- Podatkovna baza MySQL,

Za nalaganje Mobilnih aplikacij na mobilne naprave poznamo več različnih načinov namestitve:

- Nameščanje preko serijskega kabla,
- Preko Bluetooth povezave,
- Preko MMS,
- OTA (ang. Over the Air).

Pri nameščanju je predvsem pomembno, da se uporabita tako JAR kot tudi JAD datoteka. JAD je majhna tekstovna datoteka, ki nosi informacije o vsebini in tudi url naslovu kjer se paket JAR nahaja. Uporabnik ponavadi na mobilno napravo najprej naloži datoteko JAD, nato pa se na podlagi vsebine in velikosti datoteke JAR odloči, kam bo aplikacijo namestil. OTA način nalaganja aplikacije poteka tako, da je aplikacija dostopna na nekem url naslovu. Uporabnik si jo preko brskalnika prenese na svojo mobilno napravo in jo namesti.

7. Sklepne ugotovitve

7.1 Ocena dela

V sklopu diplomskega dela je bila razvita aplikacija, ki predstavlja enega izmed možnih načinov uporabe tehnologije NFC. Diplomsko delo obsega pregled uporabljenih brezžičnih tehnologij in orodij, ki sem jih potreboval pri razvoju aplikacije. Na začetku sem se seznanil z lastnostmi in načini delovanja tehnologije RFID, nato pa še z njej sorodno tehnologijo NFC, ki je bila uporabljena za zajem podatkov iz RFID značk, v aplikaciji namenjeni za interaktiven ogled razstav. Poleg NFC in RFID tehnologij sem se podrobneje seznanil z lastnostmi in načinom delovanja tehnologije Bluetooth, ki je bila v aplikaciji uporabljena kot povezovalni člen med mobilno napravo in osebnim računalnikom. Pridobil sem znanja o razvoju mobilnih aplikacij v programskem jeziku Java ME in znanja o razvoju aplikacij odjemalec/strežnik, ki kot način komunikacije uporabljajo tehnologijo Bluetooth. Dopolnil sem znanje iz programskega jezika Java SE in spoznal nekatera nova orodja, ki mi bodo tudi v prihodnje koristila pri razvoju mobilnih in namiznih aplikacij.

7.2 Ideje in funkcionalnosti za dopolnitev aplikacij

Razširitev aplikacije odjemalec strežnik:

- Na vhodu bi si uporabnik na napravo prenesel zemljevid muzeja v digitalni obliki, lahko pa bi bil zemljevid tudi del aplikacije.
- Mobilna aplikacija bi se povezala z GPS sistemom in obiskovalcu pokazala, kje na zemljevidu muzeja se v določenem trenutku nahaja.
- Ko bodo na voljo naprave z NFC in Wi-Fi podporo, bi komunikacija med napravami lahko potekala preko Wi-Fi.
- Prenos avdio in video posnetkov, o razstavnih eksponatih.
- Avdio, video posnetke in tekst bi predvajali v jeziku, ki si ga uporabnik izbere.
- Lahko bi imeli posebne značke, za skupinske obiske. Informacije o eksponatih, bi se prenesle na LCD zaslon ali avdio napravo, ki bi začela predvajati video, avdio posnetke v prostoru.
- Uporabnik bi lahko izbral zahtevnost prikazane vsebine, glede na svoje predznanje (brez predznanaja (osnovna predstavitev), s predznanjem (zahtevnejše)).
- Uporabnik bi lahko izbral način interakcije (način v obliki kviza, raziskovalni način, način za otroke).

Razširitev statistike:

- Lahko bi izračunali povprečni čas obiska muzeja, ki bi ga navedli obiskovalcem.
- Obiskovalcem bi lahko predstavili najpogostejšo pot.
- Izdelal bi se poročilni sistem (ang. reporting services).

Razširitev spletne strani:

- Nakup karte bi se lahko opravil preko mobilnega telefona ali preko spletne strani.
- Obiskovalec bi na začetku v aplikacijo vpisal svoj elektronski naslov, da bi mu bilo posredovano uporabniško ime in geslo, s katerim bi se prijavil na spletno stran muzeja. Na spletni strani bi podoživel izkušnjo, ki jo je bil deležen v muzeju. Lahko bi si ogledal eksponate katere si je ogledal na poti skozi muzej ali pa tudi vse eksponate (Geslo bi imelo omejen čas delovanja).

Dodatek A

Seznam slik

Slika 1. Frekvence RFID.	7
Slika 2. Čitalec in značka RFID.	8
Slika 3. Načina sklopitev RFID čitalnika in značke.	9
Slika 4. Možne oblike RFID značk, glede na način napajanja.	10
Slika 5. Model induktivne sklopitve med čitalnikom in oznako.	13
Slika 6. NFC telefon Benq T80 in RFID značka.	14
Slika 7. Bluetooth sklad in plasti.	16
Slika 8. Prikaz Bluetooth omrežja.	18
Slika 9. Grafični prikaz delitve Java 2 platforme.	21
Slika 10. Platforma Java2 ME.	23
Slika 11. Nadzorna plošča XAMPP namenjena zagonu strežnikov Apache in MySQL.	27
Slika 12. Možnosti uporabe Bluesoleil + Bluetooth USB Adapterja.	30
Slika 13. Desktop PC + USB Bluetooth adapter.	31
Slika 14. Nokia 6212 Classic.	32
Slika 15. Block diagram značke Mifare Classic.	36
Slika 16. Pritličje muzeja NFC-TM.	37
Slika 17. Prvo nadstropje muzeja NFC-TM.	38
Slika 18. Orodje za administracijo podatkovnih baz MySQL.	39
Slika 19. Struktura tabel v podatkovni bazi tehničnega muzeja.	39
Slika 20. Tabela razstavnih eksponatov.	40
Slika 21. Umestitev Bluecove knjižnice v Bluetooth sklad.	41
Slika 22. Komunikacija med Bluetooth odjemalci in strežnikom.	47
Slika 23. Življenski cikel MIDlet aplikacije.	48
Slika 24. Načini komunikacije med napravami.	55
Slika 25. Obrazec za vnos novih razstavnih eksponatov.	57
Slika 26. Obrazec za vnos novih oddelkov oziroma razstav.	58
Slika 27. Obrazec za nastavitve statusa posameznim oddelkom.	58
Slika 28. Forma za pregled statistike.	59
Slika 29. Komunikacije med odjemalcem, strežnikom in podatkovno bazo.	60
Slika 30. Spletna stran tehničnega muzeja NFC-TM.	61
Slika 31. Stran, ki prikazuje oddelek za mobilno telefonijo.	62

Dodatek B

Seznam tabel

Tabela 1. Bluetooth profili.....	17
Tabela 2. Energijski razredi Bluetooth naprav.....	19
Tabela 3. Podmape, ki jih vsebuje paket JDK.	22
Tabela 4. Parametri naslova URL.....	42
Tabela 5. Paketi, ki jih vsebuje vmesnik JSR 257.	48
Tabela 6. Tabela razredov in vmesnikov, ki so vsebovani v paketu javax.microedition.contactless.	49
Tabela 7. Parametri metode <i>searchServices</i>	51

8. Seznam uporabljenih virov

- [1] V. D. Hunt, A. Puglia, M. Puglia, RFID – A guide to radio frequency identification, New Jersey, Wiley, 2007, uvod
- [2] (2009) Skupina RFID. Dostopno na:
<http://www.skupinarfid.com>
- [3] C. Reinhold, P. Scholz, W. John in U. Hilleringmann, Journal of Communications vol.2, No.6 , 2007
- [4] (2007) NFC News. Dostopno na:
<http://www.nfcnews.com/2007/11/12/nfc-in-action-at-cartes-2007>
- [5] (2009) Bluetooth. Dostopno na:
<http://www.Bluetooth.org>
- [6] (2009) Wiki Forum Nokia. Dostopno na:
http://wiki.forum.nokia.com/index.php/Bluetooth_Protocol
- [7] U.Mesojedec Java 2: temelji programiranja, Ljubljana, Pasadena, 2004, I.del.
- [8] (2009) Eclipse. Dostopno na:
<http://www.eclipse.org/>
- [9] (2009) Sun. Dostopno na:
<http://java.sun.com/products/>
- [10] (2009) Softpedia. Dostopno na:
<http://www.softpedia.com/>
- [11] (2009) Apache Friends. Dostopno na:
<http://www.apachefriends.org/en/xampp-windows.html>
- [12] (2009) MySQL. Dostopno na:
<http://www.mysql.com/about/>
- [13] (2009) PhpMyAdmin. Dostopno na:
http://www.phpmyadmin.net/home_page/index.php
- [14] (2009) Nokia. Dostopno na:
<http://www.nokia.com>
- [15] (2009) Bluesoleil. Dostopno na:
<http://www.bluesoleil.com/products/Default.aspx>
- [16] (2009) Mifare. Dostopno na:
<http://mifare.net/>

[17] (2009) Mifare varnost. Dostopno na:
<http://mifare.net/security>

[18] (2009) Bluecove. Dostopno na:
<http://bluecove.org/>

[19] (2009) MySQL Connector. Dostopno na:
<http://www.mysql.com/products/connector/>