



UNIVERZITET SINGIDUNUM
DEPARTMAN ZA POSTDIPLOMSKE STUDIJE
MASTER STUDIJE

MASTER RAD

ANDROID APLIKACIJA ZA HEALTH CARE SISTEM

Mentor:
Prof. dr Ranko Popović

Student:
Branka Rodić 410608/2011

Beograd, 2012. godine

Sadržaj

1	Uvod.....	5
2	METODOLOGIJA ISTRAŽIVAČKOG RADA.....	7
2.1	Predmet istraživanja.....	7
2.2	Cilj istraživanja	7
2.3	Metode istraživanja	7
3	Arhitektura sistema eHealth i mHealth	8
3.1	Definicija eHealth (“e-Zdravlja” ili “e-zdravstva”)	8
3.2	Izgradnja eHealth u Srbiji	9
3.3	mHealth (“m-Zdravlje”) kao deo eHealth	13
3.4	Definicija mHealth	13
3.5	mHealth u službi unapređenja zdravlja	14
3.6	mHealth u Srbiji.....	15
3.7	mHealth aplikacije	16
3.8	Postojeće mHealth aplikacije i dosadašnja istraživanja	17
3.8.1	Mobilni sistem zakazivanja	18
3.8.2	Aplikacije za praćenje i kontrolu lekova	19
3.8.3	Aplikacije za udaljenu dijagnostiku	20
3.8.4	Aplikacije za poboljšanje kvaliteta života (well being).....	21
4	Mobilni uređaji i njihove karakteristike.....	23
4.1	Operativni sistemi pametnih telefona	24
4.1.1	Android	25
4.1.2	iPhone	27
4.1.3	Symbian operativni sistem.....	28
4.1.4	Windows Mobile	28
4.1.5	Windows phone	29
4.1.6	BlackBerry	30
4.2	Zajedničke karakteristike mobilnih uređaja	30
5	Dostupne tehnologije i alati za implementaciju mHealth sistema.....	31
5.1	XML (Extensible Markup Language)	31

5.1.1	Pojam XML-a.....	31
5.1.2	Karakteristike XML-a	31
5.1.3	Upotreba XML-a	32
5.2	JSON (JavaScript Object Notation).....	33
5.3	Web servisi.....	34
5.3.1	Pojam Web servisa.....	34
5.3.2	Arhitektura Web servisa.....	35
5.4	SOAP (Simple Object Access Protocol)	37
5.5	WSDL – Web Service Description Language.....	37
5.6	REST (Representational state transfer)	38
5.6.1	EksPLICITNO korišćenje HTTP metoda	39
5.6.2	Stateless (bez stanja)	39
5.6.3	URI sa strukturom direktorijuma	39
5.6.4	Prenos XML, JSON ili oba	40
5.7	WCF - Windows Communication Foundation.....	40
5.7.1	Rešavanje problema: šta WCF obebeđuje	40
5.7.2	Osnovni koncepti WCF-a.....	45
5.7.3	Razlike između WCF i Web servisa.....	46
5.8	Standardi i okviri za razvoj web servisa	47
5.8.1	J2EE i Microsoft.NET pristup web servisima	47
5.8.2	Komparacija J2EE i .NET platforme	52
5.9	Programska okruženja za razvoj web servisa	53
5.9.1	.NET proizvodi i servisi	53
5.9.2	Eclipse	54
5.10	Android i programska okruženja za razvoj Android aplikacija	55
5.10.1	Android	55
5.10.2	Programski jezik Java	55
5.10.3	Sigurnost izvršenja Java koda.....	56
5.10.4	Kompajliranje Java koda	56
5.10.5	Programska okruženja za razvoj Android aplikacija	57
6	Realizovano rešenje.....	58
6.1	Servisna podrška Android aplikaciji.....	58
6.1.1	Implementacija WCF servisa	58

6.2	Realizacija Android aplikacije.....	63
7	Zaključak.....	71
8	Prilozi	73
8.1	Upitnik za davaoca – lični podaci o davaocu.....	73
8.2	Obrazac sa pitanjima i saglasnost.....	74
9	Literatura.....	75

1 Uvod

Danas je gotovo nezamislivo poslovanje, odnosno pružanje kvalitetnih zdravstvenih informacija i usluga i razmena informacija između subjekata u zdravstvu bez savremenih informacionih tehnologija i veza.

Do skoro, komunikacija se u potpunosti odvijala u papirnoj formi, veliki deo efektivnog radnog vremena zdravstvenih radnika je korišćen na ispunjavanje obrazaca, dnevnih, mesečnih i drugih izveštaja.

Usled različitih faktora, od nedovoljno finansijskih sredstava, nepokretnosti, neupućenosti, do nepogodnih vremenskih uslova koji onemogućavaju pristup ustanovama ili isuviše dugih redova čekanja, pojedincima je onemogućen pristup ustanovama za pružanje zdravstvenih usluga ili edukacije u svrhu očuvanja i unapređenja zdravlja. U takvim situacijama korisnici zdravstvenih usluga ostaju uskraćeni za neophodnu zdravstvenu uslugu, savet ili informaciju.

Informaciono-komunikacione tehnologije (IKT) imaju suštinsku ulogu u svakodnevici današnjeg digitalnog društva. IKT su zastupljene u svim sferama života, a važnu ulogu imaju kod boljeg i efikasnijeg pružanja zdravstvenih usluga.

U radu je prikazan uticaj modernih informaciono-komunikacionih tehnologija i mobilnih uređaja na razvoj i unapređenje zdravstvene nege i prevencije zdravlja.

eHealth¹ kao pojam koji označava korišćenje informacija i komunikacionih tehnologija u zdravstvenom sektoru, se razvio iz potrebe brže i bolje identifikacije korisnika zdravstvene zaštite, pružanja kvalitetnijih i sveobuhvatnijih zdravstvenih usluga i sprovođenja preventivnih mera, upravljanja sve obimnijom dokumentacijom i informacijama i mnogih drugih razloga.

Zahvaljujući eHealth, doktori su u prilici da mnogo efikasnije pretražuju pacijentov zdravstveni karton; dođu do rezultata laboratorijskih testova u najkraćem roku, kao i da dostave potrebne recepte direktno farmaceutskoj ustanovi i dr.

Na nivou EU uvođenje eHealth označava lakši pristup zdravstvenim uslugama, bez obzira na geografsku lokaciju, a sve to zahvaljujući inovativnim rešenjima kao što su telemedicina i personalni zdravstveni sistemi. eHealth ruši postojeće barijere omogućavajući pružaocima zdravstvenih usluga (javnih ustanova, bolnica i sl.) iz svih zemalja članica da rade zajedno. Ako se neki određeni tretman za pacijenta može primeniti efikasnije u nekoj drugoj zemlji članici, onda eHealth sistem omogućava da se jednostavnije organizuje i sprovede lečenje u drugoj zemlji.

U poslednjih nekoliko godina zabeležen je ogroman porast broja mobilnih uređaja kod stanovništva zemalja u razvoju. Istovremeno se beleže loši uslovi za pružanje zdravstvenih usluga. Takva situacija je pokrenula razvoj novog pravca u okviru eHealth, tzv. mHealth. mHealth ili mobilno zdravlje je pojam koji se koristi kod

¹ Pojam označava elektronsko zdravlje, a prevodi se i upotrebljava kao e-zdravlje ili e-zdravstvo

pružanja zdravstvenih i medicinskih usluga i u javnom zdravlju, a uz podršku mobilnih uređaja.

Na osnovu istraživanja zdravstvenog stanja u svetu proistekli su i mnogobrojni programi u svetu, a koji imaju za cilj implementaciju mHealth kako bi se poboljšao kvalitet pružanja zdravstvenih usluga, edukacija stanovništva, smanjile stope bolesti i smrtnosti i ostalo.

U radu smo dali pregled trenutnog stanja i plana za budući razvoj informacionog sistema u zdravstvu u Srbiji i realizovali smo Android aplikaciju, na osnovu originalnog upitnika za dobrovoljne davaoce krvi, koja omogućava dobrovoljnim davaocima krvi da na jednostavan način popune upitnik i registruju se, a potom izvrše validaciju odgovora na pitanja o bezbednosti dobrovoljnog davanja krvi.

2 METODOLOGIJA ISTRAŽIVAČKOG RADA

2.1 Predmet istraživanja

Predmet istraživanja rada su savremeni pojavni oblici zdravstva i informatike, ujedinjeni i uobličeni u pojmove eHealth i mHealth.

U drugom i trećem delu rada dat je pregled i analiza modernih raspoloživih tehnologija i alata kojima je moguće implementirati mHealth aplikaciju i servisnu podršku za aplikaciju.

U poslednjem delu rada prikazana je Android aplikacija koju smo realizovali za unapređenje zdravstvenog stanja i postizanja benefita putem mobilnog uređaja.

2.2 Cilj istraživanja

Cilj rada je realizacija Android aplikacije za zdravstveni sistem, a da bi se postigao zadati cilj neophodno je sagledavanje i upoznavanje sa postojećim stanjem informatike u zdravstvu, preko pojmova eHealth i mHealth i budućim pravcima razvoja eHealth i mHealth kod nas i u svetu.

2.3 Metode istraživanja

Istraživanje je utemeljeno na osnovama domaće i strane naučne i stručne literature. U toku istraživanja korišćeni su osnovni principi naučnog saznanja: objektivnost, pouzdanost, opštost i sistematičnost.

Metode koje će biti korišćene u radu temelje se na dostupnoj naučnoj i stručnoj literaturi, zakonskim aktima i uredbama koje se odnose na zdravstveni sistem u našoj zemlji.

Na osnovu postavljenog cilja istraživanja izabrane su metode analize i sinteze, metoda klasifikacije, metoda indukcije i dedukcije, kao i tablično i grafičko prikazivanje statističkih podataka.

Metoda komparacije se koristi u drugom i trećem delu rada, a rezultat komparacije jeste izabrana savremena informaciona tehnologija koja će biti korišćena za realizaciju mHealth aplikacije.

3 Arhitektura sistema eHealth i mHealth

3.1 Definicija eHealth (“e-Zdravlja” ili “e-zdravstva”)

Prema definiciji Svetske zdravstvene organizacije, eHealth označava korišćenje informacija i komunikacione tehnologije u zdravstvenom sektoru [1].

eHealth je nastao kao područje gde se ukršta medicinska informatika, javno zdravlje i procesi koji se tiču pružanja zdravstvenih usluga i informacija koje se generišu i prenose preko interneta i pripadajućih tehnologija. U širem smislu ovaj termin označava ne samo tehnički razvoj, već i promene načina razmišljanja, sticanje pozitivnih navika, umrežavanje, a sve u cilju poboljšanja zdravstvene nege na lokalnom, regionalnom i globalnom nivou, korišćenjem informacija i komunikacionih tehnologija [2].

Prema strategiji razvoja informacionog društva u Republici Srbiji iz 2005. godine, eHealth označava korišćenje i modernih IKT da bi se udovoljilo potrebama građana, korisnika zdravstvene zaštite, zdravstvenih stručnjaka, ustanova koje pružaju zdravstvene usluge i kreatora zdravstvene politike.

Kao jedna od standardnih IKT aplikacija u okviru nacionalne strategije za razvoj informacionog društva, eHealth doprinosi održivom razvoju i donosi dobrobit u svim aspektima života našeg stanovništva. eHealth korenito menja pristup zdravstvenim informacijama, jer ne koristi papir kao medijum, nego se sve informacije o korisniku i njegovom stanju zapisuju elektronski, što olakšava pristup tim podacima, bilo lokalno ili putem interneta. Podaci su dostupni bez obzira na to gde se nalaze korisnici i zdravstveni radnici ili pak gde se podaci čuvaju [3].

eHealth obuhvata širok spektar usluga ili sistema, a koji imaju dodirnih tačaka sa medicinom ili zdravstvom i informacionim tehnologijama [25]:

- **Elektronski zdravstveni karton:** omogućava razmenu podataka o pacijentima između različitih zdravstvenih stručnjaka;
- **Telemedicina:** pružanje fizičkih i psihičkih tretmana i usluga na daljinu;
- **Uslužna medicinska informatika:** korišćenje elektronskih medicinskih resursa od strane zdravih ili bolesnih ljudi;
- **Zdravstveno naučni menadžment:** u smislu mogućnosti pristupa najnovijim medicinskim časopisima, vodičima dobre prakse ili epidemiološkim činjenicama, npr. pristupi lekarskim bazama Medscape i MDLinx;
- **Virtuelni medicinski timovi:** sačinjeni od medicinskih stručnjaka koji sarađuju i vrše razmenu informacija o pacijentu preko digitalne opreme;
- **mHealth ili m-zdravlje** koje uključuje korišćenje mobilnih uređaja za prikupljanje relevantnih zdravstvenih informacija, pružanje medicinskih informacija stručnjacima iz raznih medicinskih oblasti, istraživačima, pacijentima i sl., praćenje vitalnih znakova kod pacijenata u realnom vremenu ili direktno pružanje zdravstvenih usluga;

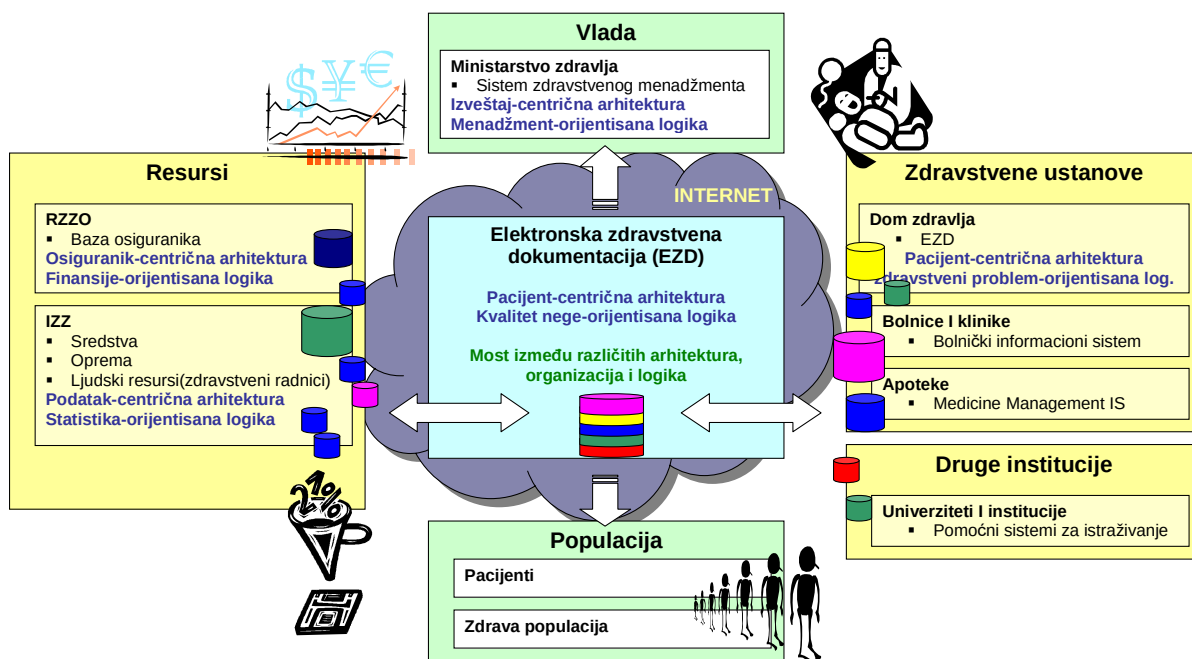
- **Medicinsko istraživanje korišćenjem Gridova:** moćno upravljanje podacima sa mogućnošću upravljanja velikom količinom različitih podataka;
- **Zdravstveni informacioni sistem:** često poistovećen sa softverskim rešenjima za zakazivanje pregleda, upravljanje podacima o pacijentu, upravljanje radnim vremenom i drugim administrativnim poslovima.

3.2 Izgradnja eHealth u Srbiji

U 2005. godini započela je izgradnja projekta „Razvoj zdravstvenog informacionog sistema za zdravstvene i farmaceutske usluge u Srbiji” Ministarstva zdravlja u saradnji Euro Health Group iz Danske [7].

Uzimajući u obzir specifičnosti zdravstvenog sistema u Srbiji, u projektu se identifikuju najvažnije komponente kroz koje treba da se operacionalizuju osnovni “building blocks” e-zdravlja.

Njihov okvir i sadržaj su diktirani i usklađeni primarno sa organizacionim aspektima srpskog zdravstvenog sistema, a ne tehnološkim resursima.



Slika 1. Elektronska zdravstvena dokumentacija (EZD) unutar zdravstvenog sistema u Srbiji

Jedna od osnovnih komponenti je Sistem elektronske zdravstvene dokumentacije (EZD) koji se bazira na elektronskoj zdravstvenoj dokumentaciji pojedinačnog korisnika zdravstvene zaštite objedinjenoj preko jedinstvenog identifikatora kroz ceo zdravstveni sistem.

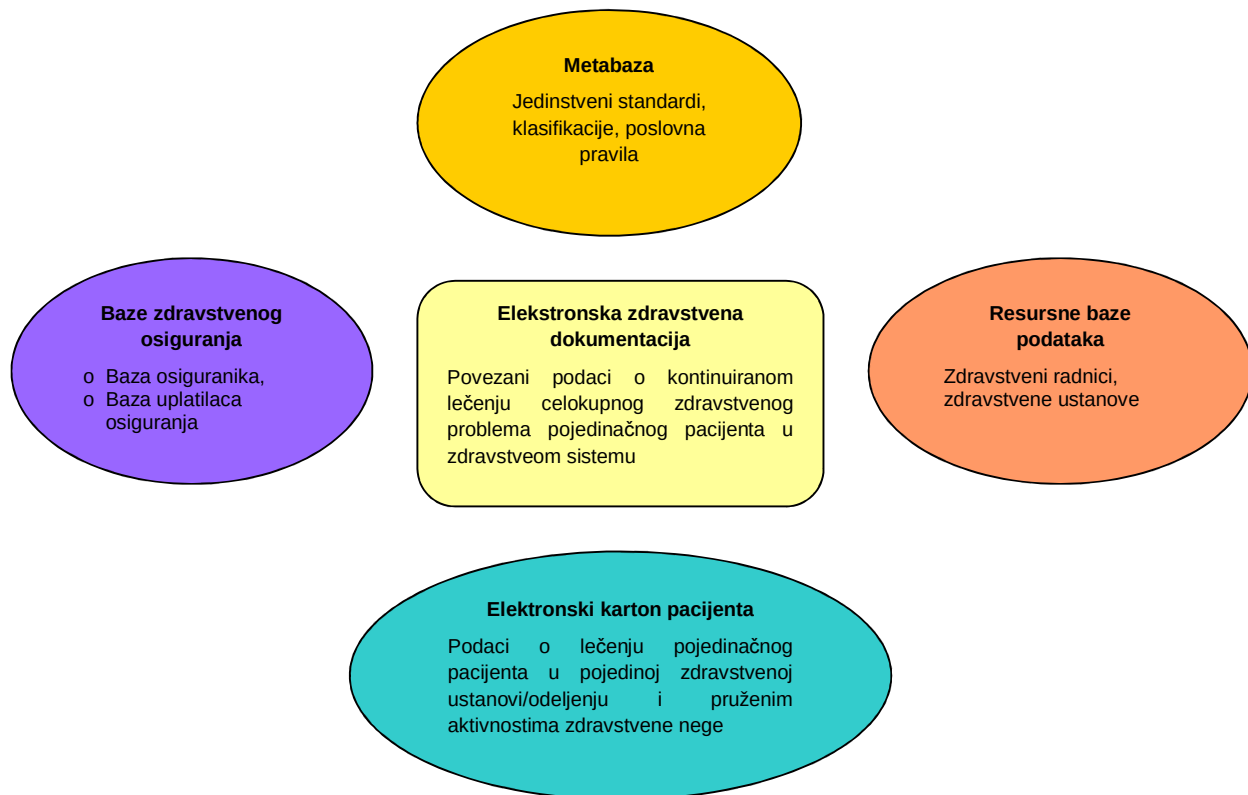
EZD je sistem koji pruža podatke o osnovnim procesima u zdravstvenom sistemu vezanim za zdravstvenu negu pojedinačnih korisnika zdravstvene zaštite. Kroz sistem EZD odabrani podaci iz raznih izvora (različitih institucija i lokacija) su sakupljeni, uskladišteni i dostupni svim autorizovanim korisnicima. Isti podaci mogu se videti na razne načine: pacijent-centrični za doktore, agregirani za statističare, analitički obrađeni za donosiocje odluka (menadžere), multidimenziono korelirani za istraživače, informativni za široku javnost, itd. [7].

Osnovna uloga sistema EZD je da utre put novom pristupu obavljanju poslovnih procesa u zdravstvenom sistemu podržanom informaciono-komunikacionim tehnologijama. Taj novi pristup karakteriše okrenutost pacijentu, medicina zasnovana na dokazima, razmena informacija o zdravlju pacijenta u cilju poboljšanja zdravstvenih usluga i smanjenju troškova.

Operativna uloga sistema EZD je da:

- uspostavi zajedničke standarde na kojima se zasniva razmena podataka u zdravstvenom sistemu;
- obezbedi minimalan skup podataka o lečenju zdravstvenog problema pojedinačnog pacijenta;
- obezbedi minimalan skup podataka o zdravstvenom stanju pojedinačnog pacijenta;
- izgradi infostrukturu za upravljanje ovim podacima (prikupljanje, uskladištenje, distribucija);
- obezbedi ovlašćeni pristup tim podacima sa raznih lokacija, nezavisno od mesta gde se oni stvarno nalaze;
- obezbedi visok stepen privatnosti i sigurnosti.

Sistem EZD u okviru globalnog zdravstvenog informacionog sistema može se takođe predstaviti i ovako sa gledišta podataka:



Slika 2. Sistem EZD sa gledišta podataka

Sistem EZD je integrativni faktor za celokupan zdravstveni informacioni sistem. Njegovo uvođenje bi trebalo da ubrza uređenu (organizovanu i sinhronizovanu) kompjuterizaciju zdravstvenih ustanova, u skladu sa standardima i usvojenim pravilima. Zajednički elementi integracije obuhvataju:

- standarde arhitekture,
- minimalni skup podataka za razmenu,
- zajedničke sisteme identifikacije i kodiranja,
- pravila sigurnosti i privatnosti,
- XML šeme za razmenu poruka.

U 2009. godini u Srbiji je doneta Uredba [8] kojom se utvrđuje program rada, razvoja i organizacije integrisanog zdravstvenog informacionog sistema "e-zdravlje".

Programom se utvrđuju okvir, principi, ciljevi i način upravljanja integrisanim zdravstvenim informacionim sistemom, kao i aktivnosti, nosioci aktivnosti, izvori finansiranja i rokovi za izvršavanje ciljeva Programa za period 2009-2015. godine [8].

Princip ovog programa jeste da informaciono komunikacione tehnologije (IKT) budu

sredstvo za poboljšanje efikasnosti pristupa informacijama, međusobnoj komunikaciji, boljem praćenju kvaliteta i boljem planiranju.

Primena IKT mora da omogućiti:

1. Zdravstvene informacije koje će da pomognu zdravstvenim radnicima u donošenju kliničkih odluka, uključujući i vodiče dobre prakse, baze znanja i stručnu literaturu.
2. Upotrebu podataka u cilju podrške funkcijama javnog zdravlja, planiranju, nadgledanju i oceni zdravstvenih usluga, uključujući upravljanje i planiranje kadrovima, finansiranju i alokaciji resursa, praćenju kvaliteta zdravstvenih usluga, i praćenju racionalnog trošenja sredstava.
3. Upotrebu podataka kao podrške razvoju i primeni odluka u cilju sveobuhvatne zdravstvene zaštite pojedinaca, grupa sa posebnim potrebama i cele populacije.
4. Upotrebu podataka u cilju pomoći zdravstvenim istraživanjima.
5. Informacije o stanju zdravlja i zdravstvenim determinantama, zasnovanim na dokazima.
6. Lakše ispunjavanje međunarodnih obaveza kroz usvajanje evropskih standarda i inicijativa.

Principi koje primena IKT mora da zadovolji jesu:

- očuvanje privatnosti i poverljivosti ličnih zdravstvenih podataka;
- efikasnost i upotrebljivost zdravstvenog informacionog sistema;
- promociju optimalne upotrebe zdravstvenih podataka;
- visok kvalitet zdravstvenih informacija.

Cilj Programa jeste da se do 2015. godine upotrebom informacionih i komunikacionih tehnologija omogući nesmetano i kvalitetno funkcionisanje svih delova sistema zdravstvene zaštite, kroz:

- automatizaciju i smanjenje troškova svih administrativnih postupaka i procesa koji prate osnovne delatnosti sistema zdravstvene zaštite;
- blagovremen prihvatanje podataka i podršku mogućim izmenama i proširenjima;
- sigurnu i efikasnu razmenu informacija između svih učesnika zdravstvenog sistema, u cilju podizanja dostupnosti i kvaliteta zdravstvene zaštite;
- aktivno učešće građana u brizi o sopstvenom zdravlju, pre svega u smislu potpune informisanosti i određene slobode izbora, stepena odlučivanja i uticaja na sopstveni tretman, kao i učešća u prevenciji;
- formiranje elektronske baze znanja zdravstvenog sektora;
- razmenu informacija od značaja za obavljanje naučno istraživačke delatnosti, kao i obavljanje permanentnog obrazovanja medicinskog osoblja.

Do opštih ciljeva dolazi se preko operativnih ciljeva, a to su:

- ostvarivanje preduslova za efikasan IKT razvoj;
- elektronske evidencije;
- pojedinačni elektronski servisi;
- informatička pismenost i veštine zdravstvenih radnika;
- zapošljavanje IKT stručnjaka [8].

Kao prioritetni posebno se izdvajaju pojedinačni elektronski servisi koji treba da obezbede:

1. prikaz zdravstvenih ustanova (državni i privatni sektor) sa zdravstvenim uslugama koje mogu da ponude građanima;
2. zakazivanje pregleda;
3. interaktivne zdravstvene savete [8].

3.3 mHealth (“m-Zdravlje”) kao deo eHealth

Pojava i širenje web-a su revolucionarno uticali i promenili način interakcije, prikupljanja i distribucije informacija.

Do nedavno, informacije su bile dostupne samo korisnicima desktop računara. Sa izumom mobilnih telefona sa pristupom internetu omogućeno je globalno širenje informacija.

Tržište mobilnih i smart telefona je toliko napredovalo da i najjeftiniji mobilni telefon ima neki od načina pristupa internetu.

Sa niskom cenom i ubrzanom penetracijom mobilnih telefona u svetu, na desetine miliona ljudi, koji ranije nisu imali pristup fiksnom telefonu ili računaru, sada koriste mobilni uređaj kao svakodnevni alat za komunikaciju i prenos podataka.

Mobilna komunikacija daje nove mogućnosti širenja zdravstvenih informacija među stanovništvom zemalja u razvoju, a njihova sveprisutnost je glavni razlog zbog kojeg oni treba da budu zastupljeni u novim tehnološkim rešenjima za pružanje zdravstvenih usluga.

Ruralne i marginalizovane oblasti oskudevaju u infrastrukturi, a pomoć u vidu mobilne zdravstvene usluge može da promeni njihov način života [25].

3.4 Definicija mHealth

Pojam mHealth je proistekao iz pojma eHealth, odnosno kao njegov segment.

mHealth ili mobilno zdravlje je pojam koji se koristi kod pružanja zdravstvenih i medicinskih usluga i u javnom zdravlju, a uz podršku mobilnih uređaja. Tu se misli na mobilne komunikacione uređaje, kao što su mobilni telefoni, tablet računari, PDA uređaji, a koji se koriste u svrhu dobijanja ili pružanja zdravstvenih servisa ili informacija [16].

Do danas nije ustanovljena standardizovana definicija mHealth. Za potrebe izveštavanja, The Global Observatory for eHealth (GOe) je definisala mHealth ili mobilno zdravlje kao medicinsku praksu i delovanje u sferi javnog zdravlja podržano mobilnim uređajima kao što su mobilni telefoni, uređaji za praćenje stanja pacijenta, Personal Digital Assistants (PDAs) i drugi bežični uređaji [16].

mHealth uključuje upotrebu i korišćenje osnovnih servisa mobilnih telefona kao što su glasovni prenos, servis kratkih poruka (SMS), isto kao i mnogo složenije funkcije i aplikacije uključujući bežične tehnologije kao što su Bluetooth, GPRS, treću i četvrtu generaciju mobilnih telekomunikacija (3G i 4G sistemi), WiFi, WiMAX i sl. za prenos različitih eHealth podataka i servisa.

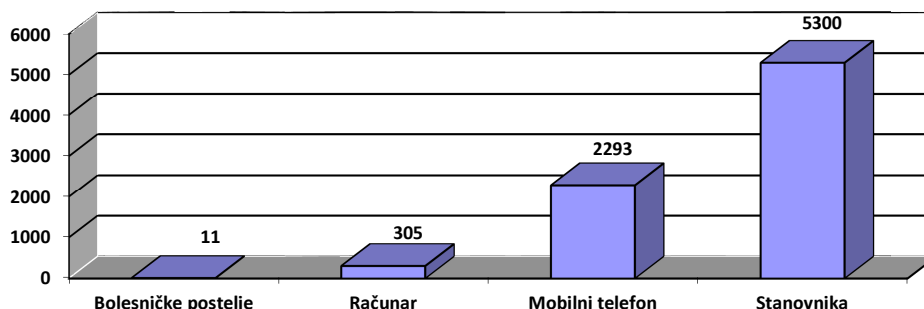
3.5 mHealth u službi unapređenja zdravlja

Ljudi u ruralnim sredinama nemaju u svojoj blizini odgovarajući pristup kvalitetnoj zdravstvenoj zaštiti. Iako postoje zdravstveni centri ili stanice u selima, oni uglavnom nisu snabdeveni dovoljnim brojem osoblja, a najviše im nedostaje osnovna infrastruktura. Stanovništvo tih sredina je prinuđeno da prelazi velike razdaljine do najbližih opština ili većih gradova u regionu kako bi dobili medicinske usluge, često toliko skupe da ih ne mogu sebi priuštiti [4].

Sa druge strane, pružaoci zdravstvenih usluga se suočavaju sa sve većim izazovima u vezi sa dijagnostikom i prikupljanjem informacija usled neadekvatne infrastrukture i manjka zdravstvenih radnika koji bi sprovodili takve usluge.

Tradicionalan način prikupljanja informacija se bazira na zapisima, samoprocenama i neredovnim posetama lekaru [5].

Prema Terry Kramer, direktoru za strateški razvoj Vodafon-a, postoji 2,2 milijarde mobilnih telefona u zemljama u razvoju, 305 miliona kompjutera i samo 11 miliona bolesničkih postelja [26].



Slika 3. Statistički prikaz zastupljenosti tehnologija i zdravstvene infrastrukture u zemljama u razvoju (u milionima)

Ovo jasno pokazuje kolika moć leži u mobilnim tehnologijama i kako se one mogu iskoristiti kao snažno i jeftino oruđe za prevazilaženje problema nedostataka zdravstvenih resursa u zemljama u razvoju.

mHealth kao oblast počiva na premisi da tehnologija integrisana unutar zdravstvenog sektora ima ogroman potencijal da uspostavi bolju zdravstvenu komunikaciju, promoviše zdrave stilove života, olakša procese odlučivanja, kako za

stručnjake tako i za pacijente, i konačno da poboljša kvalitet zdravstvenih usluga omogućavajući lakši pristup medicinskim i zdravstvenim informacijama u onim mestima gde je pristup informacijama ranije bio otežan ili nemoguć.

Bez obzira na skorašnju pojavu i definiciju mHealth, ova oblast je uveliko uticala na način pružanja medicinskih i zdravstvenih usluga i informacija u svetu. Mnogi projekti sprovedeni u zemljama u razvoju ističu konkretne benefite nakon primene inovacija iz ove oblasti [6]:

- povećan je pristup zdravstvenim uslugama i informacijama, pogotovo nedostižnim delovima populacije,
- poboljšana mogućnost za dijagnostiku i praćenje bolesti,
- pravovremene informacije iz oblasti javnog zdravlja,
- omogućen je lakši pristup savremenoj medicinskoj edukaciji i obuci za zdravstvene radnike.

3.6 mHealth u Srbiji

Prema Uredbi kojom se utvrđuje program rada, razvoja i organizacije integrisanog zdravstvenog informacionog sistema “e-zdravlje” u Srbiji, izdvojeno je nekoliko prioriternih ciljeva koji treba da obezbede:

1. prikaz zdravstvenih ustanova (državni i privatni sektor) sa zdravstvenim uslugama koje mogu da ponude građanima;
2. zakazivanje pregleda;
3. interaktivne zdravstvene savete.

Ovo je jako dobra polazna osnova za implementaciju mHealth u Srbiji.

Do sada nisu poznati slučajevi implementacije nekog servisa uz pomoć mobilnih uređaja koji bi obezbeđivao zdravstvene informacije ili pružao zdravstvene usluge građanima Srbije, osim ako se ne radi o SMS uslugama u vidu pružanja zdravstvenih informacija ili podsetnika.

3.7 mHealth aplikacije

Mobilna aplikacija ili “mobile app” se definiše kao softverska aplikacija koja se izvršava na mobilnoj platformi, ili može da bude web aplikacija prilagođena mobilnom telefonu, a koja se izvršava na serveru [9].

Aplikacije mHealth imaju za svrhu prikupljanje kliničkih i opštih zdravstvenih informacija, obezbeđenje komunikacije između pojedinaca i zdravstvenog sistema, pružanje zdravstvenih i preventivnih informacija, monitoring i nadzor, sprovođenje preventivnih programa, praćenje zdravstvenog stanja pacijenta u realnom vremenu, kao i direktno pružanje medicinskih usluga (oblast mobilne telemedicine).

One mogu da doprinesu da se zdravstvene usluge dovedu do vulnerabilnih grupa, da se poveća efikasnost pružanja zdravstvene nege uz smanjenje troškova, kao i da se poveća efikasnost preventivnih programa, programa javnog zdravlja i istraživačkih projekata.

Aplikacije u okviru oblasti mHealth se mogu, prema UN i Vodafone fondaciji, svrstati u 6 kategorija [6]:

1. Edukacija i obrazovanje
2. SOS podrška
3. Dijagnostika i lečenje
4. Komunikacije i treninzi za zdravstvene radnike
5. Praćenje bolesti i epidemiološkog stanja
6. Daljinsko praćenje
7. Daljinsko prikupljanje podataka

Royal Tropical Institute [23] definiše osam oblasti u koje se mogu svrstati sve za mHealth aplikacije:

- *Edukativni sistemi i sistemi za podizanje svesti o zdravlju.* Sistemi koji obezbeđuju informacije o promociji zdravlja i prevenciji bolesti, edukativne programe i sl.
- *Point-of-care podrška i dijagnostika.* Koriste se da obezbede lekarima informacije o kliničkoj nezi kao i sistem za podršku odlučivanju u dijagnostici.
- *Monitoring pacijenata.* Obezbeđuju podršku lečenju i zakazivanju poseta.
- *Epidemiološki nadzor.* Obezbeđuju praćenje zaraznih bolesti u realnom vremenu.
- *Sistemi za hitne medicinske intervencije.* Obezbeđuju alarme za udese, nesreće ili prirodne katastrofe.
- *Zdravstveni informacioni sistemi.* Upravljaaju podacima koji se koriste pri lečenju.
- *eLearning (eUčenje).* Obezbeđuju mobilne platforme za podršku obrazovnim sistemima u zdravstvu ili kontinualnu edukaciju za zdravstvene radnike.
- *Aplikacije za finansije.* Olakšavaju korišćenje smart kartica ili vaučera kod mobilnih plaćanja.

Prema izveštajima o istraživanjima Univerziteta u Kembridžu i Kineskog mobilnog operatera (China Mobile) vezanih za aplikacije u mobilnoj mreži i zdravlje (mHealth aplikacije) dolazi se do saznanja da inovativne mHealth aplikacije imaju veliki potencijal da transformišu zdravstvo kako u zemljama u razvoju tako i u razvijenim zemljama.

Prema ovim izveštajima dolazimo do pet zdravstvenih oblasti u okviru kojih se radi na razvoju mHealth aplikacija:

1. Istraživanje javnog zdravlja
2. Primarna nega
3. Hitno zbrinjavanje
4. Hronična stanja
5. Plasiranje informacija i samopomoć

3.8 Postojeće mHealth aplikacije i dosadašnja istraživanja

Mobilna tehnologija je primenjena kroz mnoge inicijative u svetu.

Inicijativa The Opportunity of Mobile Technology for Healthcare in the Developing World je prepoznatljiva širom sveta i sprovodi preko 50 projekata u 26 zemalja.

Prema [6], najviše projekata mobilnih tehnologija su sprovodile Indija sa 11 projekata i Južna Afrika i Uganda sa po šest.

Neki od mobilnih projekata su:

- Slanje SMS vlasnicima mobilnih telefona o podacima o bolesti
- Omogućavanje zdravstvenim radnicima u Ugandi unos podataka preko mobilnog telefona na terenu
- SIMpill ima ugrađen senzor u flašici za pilule sa SIM karticom koja obaveštava doktora u Južnoj Africi da li je pacijent uzeo lekove za tuberkulozu
- U Ugandi je sproveden kviz o HIV/AIDS za 15000 pretplatnika. Oni koji su odgovorili na pitanja dobili su besplatne mobilne minute. Na kraju kviza, poslat je finalni SMS koji je usmeravao volontere za testiranje
- U Brazilu, zdravstveni radnici su popunjavali upitnik preko mobilnog telefona za istraživanje o slučajevima hemoragične groznice
- U Meksiku, medicinski hotline – MedicalHome omogućava pacijentima da pošalju pitanja putem SMS-a

Ovi projekti su doneli rezultate u vidu poboljšanja zdravlja pacijenata i to na više načina: uspostavljena je bolja redovnost kod režima lečenja (prilikom uzimanja lekova ili druge terapije), podignut je nivo kolektivne svesti, poboljšana je kontrola bolesti i dr. [26].

Mnoge eHealth aplikacije koje se koriste širom sveta su tehnički jednostavno izvedene (češće je korišćenje prenosa glasa i SMS nego neke naprednije funkcionalnosti), neposredno zadovoljavajući korisničke potrebe ili pak nude određene pogodnosti koje podstiču korisnika da koristi aplikaciju. Mnoge današnje

aplikacije su poboljšane verzije postojećih aplikacija, a najčešće su zastupljeni sledeći tipovi [31]:

3.8.1 Mobilni sistem zakazivanja

Koristeći pogodnosti slanja SMS-a i prenosa glasa moguće je zakazati pregled kod lekara i specijaliste. Ovakvi sistemi su široko rasprostranjeni i prihvaćeni uz značajno sniženje troškova kako za zdravstvene ustanove, pružaoce usluga, tako i za korisnike takvih usluga. Najzastupljeniji su u primarnoj zdravstvenoj zaštiti.

Ovakva funkcionalnost je često standardna komponenta kliničko-bolničkog softvera. Studije su pokazale da postoje značajni benefiti primene ovakvog sistema:

- U Melburnu, Australija, procena efikasnosti SMS podsetnika za preko 20 000 pacijenata u Royal Childrens' Hospital pokazuje da se broj otkazanih pregleda značajno smanjio kada je uveden sistem SMS podsetnika.
- iPlato SMS platforma, sistem koji se koristi u UK, za podsticanje pacijenata u ustanovama primarne zdravstvene zaštite da vrše redovne preglede dojki, izvrše vakcinaciju protiv gripa ili redovnu vakcinaciju dece.
- U Kini je izvršeno sporadično praćenje SMS podsetnika i mobilnog glasovnog podsetnika u centru za promociju zdravlja u provinciji Zhejiang. Istraživanje pokazuje da su oba metoda bila efikasna i da nema razlike u smislu efikasnosti (ima razlike u ceni SMS podsetnika koja je znatno niža od telefonskog podsetnika).

Reprezentativan je primer „12580“ sistema za zakazivanje, koji je na snazi u provinciji Guangdong u Kini. Pacijenti zovu broj 12580 sa svog mobilnog telefona i razgovaraju sa osobljem „call“ centra koje im pravi rezervaciju u skladu sa njihovim potrebama i raspoloživosti kapaciteta zdravstvenih ustanova. Agenti u „call“ centru imaju sistem koji im daje informacije o raspoloživosti kapaciteta u nekoliko obližnjih zdravstvenih ustanova.

Zakazivanje se vrši od jednog do sedam dana unapred.

Nadoknada za uslugu zakazivanja može da bude plaćena unapred preko mobilnog sistema plaćanja kako bi se zaobišli redovi čekanja u samoj bolnici.

Ne plaća se dodatna naknada za rezervisanje termina (osim cene standardnog poziva broja 12580). Nakon poziva, sistem generiše broj i prosleđuje ga korisniku putem SMS poruke. Ta šifra se unosi u terminale na ulazu u zdravstvenu ustanovu i služi za autentikaciju korisnika. Korisnik dobija odštampan bar kod koji potvrđuje prijem na relevantnu kliniku.

Postoji sistemska i identifikaciona provera kako bi se sprečile eventualne zloupotrebe (prodaja zakazanih termina).

Primena ovakvog sistema je značajna za kinesko zdravstvo koje pokriva 7 gradova u Guangdong provinciji, a to pokazuje činjenica da je sistem obradio 300 000 poziva pacijenata u preko 36 „call“ centara u vremenskom periodu od tri meseca. Ukupno, sistem je opslužio 4,25 miliona pacijenata.

Kao deo „12580“ servisa, China Mobile je razvila medicinski informacioni servis koji

obezbeđuje SMS informacije i navigacionu pomoć pacijentima koji imaju zakazane preglede, informacije o zdravstvenoj ustanovi, radnom vremenu, prevozu i politici ustanove. Obezbeđen je i e-magazin o zdravlju (uz doplatu).

Iako sama osnova sistema za zakazivanje nije primarno mobilno orijentisana (zakazivanje se može obaviti i online), korišćenje mobilne telefonije pojednostavnjuje proces time što je moguće izvršiti naplatu naknade za zakazivanje preko pretplatničkog računa za mobilni telefon ili zakazati pregled sa bilo kojeg mesta.

Mobilni sistem zakazivanja i podsetnika predstavlja veoma jednostavnu aplikaciju mobilnog telefona za podršku i unapređenje postojećeg sistema zakazivanja. Ovakav sistem može da se lako implementira, a njime se osim pružanja potrebnih usluga, značajno smanjuju troškovi zdravstvenih ustanova.

Tamo gde postoji zakonska regulativa, ovakav sistem može da se integriše u EZK i postojeći sistem naplate usluga pacijentima.

3.8.2 Aplikacije za praćenje i kontrolu lekova

To su aplikacije koje se odnose na kontrolu falsifikovanja i piraterije na putu od proizvodnje lekova do krajnjeg korisnika, kao i kontrolu porekla leka i mogu da doprinesu povećanju efikasnosti suzbijanja pojave falsifikovanja lekova, naročito u zemljama gde je nerazvijen sistem za distribuciju lekova.

Mobilna infrastruktura omogućava lakše praćenje u udaljenim regionima gde jedino mobilna mreža može da obezbedi povezivanje sa centralnom bazom podataka, kako u mestima prodaje tako i na mestima upotrebe, istovremeno štiteći korisnike i kredibilitet proizvođača lekova.

U Nigeriji, Nigerian National Agency for Food and Drug Administration and Control (NAFDAC) je pustio u probni rad Mobile Authentication Service – MAS za jedan specifičan lek – Glucophage, BIOFEM koji se koristi u lečenju dijabetesa. MAS omogućava konzumentima da provere autentičnost leka tako što ogrebu nalepnicu na pakovanju leka i dobijeni kod pošalju SMS-om do servera baze podataka. Povratna poruka potvrđuje originalnost leka.

Ova usluga je za korisnike besplatna. Tehnologiju je razvio Sproxil, a mobilnu podršku pružaju nigerijski mobilni operateri MTN, Etisalat i Globalcom.

U probnom periodu, u tri grada (Lagos, Abuja, Port Harcourt), obeleženo je 735 000 pakovanja leka, poslato 22 000 SMS-ova i opsluženo 67 612 pojedinačnih korisnika.

Studija je pokazala da su sredstva uložena u probnom periodu za ovaj sistem manja od koristi koju dobijaju farmaceutske kuće zaštitom brenda i tržišta.

U New York-u je razvijen Epothecary sistem koji koristi bar kodove na kutijama lekova. Telefoni sa kamerama i GPS sistemom omogućavaju skeniranje kodova i prepoznavanje lokacija i njihovo poređenje sa kodovima iz centralne baze podataka.

U Guangdong, Kina, je razvijen "smart drug regulation" – sistem za suzbijanje falsifikovanja i piraterije. Centralna baza podataka sadrži informacije o poreklu

lekova. Svaka kutija leka je kodirana sa RFID oznakom i 2D bar kodom. Provera može da se obavi preko centralne baze podataka, a pristupa joj se putem fiksne ili mobilne mreže. Cilj sistema je uspostavljanje regulative koju bi trebalo da usvoje vladine agencije i proizvođači lekova.

U zemljama gde može doći do prekida lanca distribucije lekova, usled izostanka sistema zaštite ili slabosti ljudskog faktora, aplikacije za kontrolu i praćenje mogu da obezbede važnu kariku u kontroli kretanja leka od mesta proizvodnje do mesta potrošnje.

3.8.3 Aplikacije za udaljenu dijagnostiku

mHealth aplikacije za daljinsku dijagnostiku se najčešće koriste u primarnoj zdravstvenoj zaštiti.

Karakteristiše ih korišćenje mobilnih uređaja i mobilne mreže, preko kojih zdravstveni radnici postavljaju dijagnostiku bez potrebe da pacijent putuje do zdravstvenog centra.

Takve aplikacije su bazirane na mehanizmima za podršku odlučivanju ili imaju pristup udaljenim bazama za podršku odlučivanju.

Neke od ovih aplikacija koriste mrežu za povezivanje zdravstvenog radnika sa specijalistima, npr. teledermoskopija, prenos slika mladeža i kožnih lezija za ekspertizu.

Aplikacije mogu da povezuju zdravstvene radnike sa specijalistima ili pacijenta direktno sa zdravstvenim radnikom preko glasa, poruka ili videa.

U razvijenim zemljama, aplikacije se koriste da povežu korisnika iz opšte populacije za zdravstvenim radnikom, kao npr. 3G doktor u Velikoj Britaniji.

Kineski China mobile razvija nekoliko projekata koji obezbeđuju daljinsku konsultaciju i dijagnostiku. U provincijama Guangdong i Guizhou, China mobile sa partnerima obezbeđuje medicinske servise i informacije preko terminala (za samoposluživanje) koji su umreženi preko fiksne ili 3G/GPRS mobilne mreže.

U okviru dva projekta, koliko se trenutno sprovodi, na stotine terminala su u upotrebi u udaljenim seoskim sredinama. Cilj im je širenje zdravstvenih informacija kao i obezbeđenje medicinskih konsultacija. Ovi servisi su trenutno besplatni kako bi što više bili prihvaćeni, a u planu je naplata pojedinih servisa.

U provinciji Guangdong, terminal "Intelligent Health Management Self-Service" (inteligentni menadžment u zdravstvu) poseduje servis koji se sastoji iz šest međusobno povezanih modula koji obuhvataju sve aspekte očuvanja zdravlja, uključujući i informacije o zdravom življenju, informacije za trudnice i majke, zaštitu zdravlja na radnom mestu i podatke o javnom zdravlju. Dva modula, od postojećih šest: "One Stop Medical Report" i "Self Service Disease Recovery" predstavljaju udaljene zdravstvene konsultacije (za mase, dok lična konsultacija nije moguća na ovaj način zbog stroge zakonske regulative).

U okviru "One Stop Medical Report" korisnici mogu da unesu vrednosti raznih analiza (osnovne analize krvi, masnoća u krvi, tumor markera i sl.). Terminal im vraća referentne vrednosti i procenu zdravlja, kao i savet za održanje i unapređenje zdravlja. Terminal može da indikuje moguću bolest i predloži bolničku dijagnostiku ili zakazivanje termina preko "12580" servisa.

"Self-Service Disease Recovery" modul omogućava pretragu informacija koje su napisali medicinski eksperti i profesionalci iz Guangdong provincije, a odnose se na preko 80 vrsta bolesti. Informacije su vezane za simptome bolesti, dijagnostiku, preventivu, tabue ili zablude, dijetetske savete i sl.

Aplikacija ima za cilj prevenciju bolesti i rano prepoznavanje simptoma bolesti.

U Guizhou terminali obezbeđuju slične servise, omogućavaju upite vezane za bolesti, porodična osiguranja i cenovnike medicinskih usluga.

Usled nedostatka zdravstvenih resursa, ljudi često nemaju priliku da diskutuju o dobijenim rezultatima analiza sa lekarima osim ako rezultati ukazuju na postojanje neke ozbiljne bolesti. Takođe, oni ljudi koji žive u seoskim sredinama često nemaju vremena ili novca da bi ponovo posetili doktora.

Postoji još jedan servis za konsultovanje u provinciji Guangdong koji je razvijen od strane partnera kineskog mobilnog operatera Yihe.

Pretplatnici na kineski mobilni medicinski link mogu da šalju pitanja vezana za zdravlje (preko seoskih terminala ili mobilnim telefonom preko SMS, MMS ili „12580“ servisa) do Yiha platforme sa 30 profesionalaca iz oblasti opšte medicine koji radnim danima odgovaraju na pitanja korisnika.

Servis se naplaćuje kao regularna konsultacija. Trenutno se obrađuje 2000 poruka dnevno, a servis ima preko 1,89 miliona registrovanih korisnika.

3.8.4 Aplikacije za poboljšanje kvaliteta života (well being)

Među najinteresantnijim mHealth aplikacijama koje su razvijene u poslednje dve do tri godine su jednostavne aplikacije koje doprinose poboljšanju kvaliteta života pojedinca.

Neke od njih pružaju informacije (pohranjene u samoj aplikaciji ili prosleđene do telefona), dok druge koriste mnoge naprednije funkcionalnosti uređaja, npr. GPS, akcelerometar i sl.

Nedavna studija u USA koja je obuhvatala 3000 pojedinaca, a osnosi se na korišćenje ovakvih aplikacija, pokazuje da 9% odraslih Amerikanaca ima ovakvu aplikaciju u svom telefonu, a njih 17% je preko mobilnog telefona zatražilo neku informaciju vezanu za zdravlje.

Aplikacija RunKeeper za iPhone je besplatna, a koristi GPS za merenje brzine i pređene distance trčanjem (računa i broj potrošenih kalorija). Podaci mogu da budu preneti i na web sajt kako bi se sačuvala istorija vežbanja.

iFitness je aplikacija za iPhone i iPodTouch, a obezbeđuje programe za treniranje

koji odgovaraju različitim režimima treniranja i različitim grupama mišića. Programi mogu da budu prilagođeni pojedincu, a omogućeno je i praćenje napredovanja i zadatih ciljeva.

iQuit za iPhone je aplikacija koja obezbeđuje različite metode za ostavljanje pušenja, a korisnik se odlučuje za odgovarajući.

Aplikacija Drinks Tracker iPhone je razvijena u Velikoj Britaniji u 2009. godini, a omogućava korisnicima da izračunaju broj popijenih "jedinica" alkohola i da beleže pijeње tokom određenog perioda, nakon čega dobijaju sumarnu povratnu informaciju.

Svaki put kada korisnik popije alkohol, klikne na ekran na odgovarajuću ikonicu na ekranu telefona. Aplikacija prati i beleži količine popijenog pića. Popijena pića mogu da se čuvaju za svaki dan posebno i prikažu kao nedeljni zbirni unos, a potom se rezultati porede sa preporukama vlade Velike Britanije.

Aplikacija je deo napora koji se sprovode da se smanji količina nasilja izazvana pod dejstvom alkohola.

Ovakve vrste aplikacija su značajne jer predstavljaju alat kojim se jako brzo dolazi do cilja kod korisnika pametnih telefona, kako u razvijenim zemljama tako i u zemljama u razvoju.

Mnogobrojne postojeće aplikacije pokazuju da korišćenje mobilne infrastrukture u oblasti zdravlja značajno može doprineti širenju informacija i promociji zdravlja, a potencijali za dalji razvoj su ogromni.

4 Mobilni uređaji i njihove karakteristike

Mobilni telefon je uređaj koji može pozivati i primati telefonske pozive preko radio linka dostupnog na širokom geografskom području. Ovo je omogućeno povezivanjem na mobilnu mrežu operatera koji obezbeđuje kontakt sa javnom mobilnom mrežom.

Osim telefoniranja, mobilni telefoni novijeg datuma podržavaju široku paletu drugih usluga kao što su SMS poruke, MMS poruke, slanje i primanje elektronske pošte, internet pristup, kratko pojasnu bežičnu komunikaciju (infrared, bluetooth) i drugih bežičnih tehnologija kao GPRS/3G/4G, zatim poslovne aplikacije, igrice, fotografisanje i dr.

Mobilni telefoni koji obezbeđuju ove i slične usluge spadaju u kategoriju “smart” telefona ili “pametnih” telefona. Razlika između pametnih telefona i klasičnih telefona nije u potpunosti jasna i nije strogo definisano šta čini razliku među njima. Jedna od najznačajnijih razlika je napredni API (Application Programming Interface) na smart telefonima za pokretanje “third-party” aplikacija koje omogućavaju bolje integrisanje sa operativnim sistemom i hardverom telefona što nije karakteristika klasičnih telefona. Obzirom da očekivanja korisnika rastu iz dana u dan, pojam “pametni telefon” se postepeno gubi, a njegove performanse postaju standardi.

Prvi mobilni telefon je predstavio Dr Martin Cooper iz Motorola-e, 1973. godine, a čija je slušalica težila oko 1 kg [11].

Prvi pametni telefon napravio je IBM, 1994. godine, pod nazivom “Simon”. Korisnici su pored, osnovne funkcije telefoniranja, imali na raspolaganju klijenta za elektronsku poštu, “pejdžer”, kalendar, kalkulator i adresar. Telefon nije posedovao tastere, već se interakcija vršila putem ekrana osjetljivog na dodir [14].

Prvi pametni telefoni su bili uređaji koji su uglavnom imali funkcionalnosti PDA uređaja i mobilnog telefona.

Današnji modeli kombinuju funkcije prenosivih medija, plejera, digitalnih kamera, džepnih video kamera i GPS navigacija. Moderni pametni telefoni pored toga, poseduju i ekrane osjetljive na dodir visoke rezolucije, internet pretraživače koji mogu pristupiti i korektno prikazati standardne internet strane, pored sajtova prilagođenih mobilnim telefonima, kao i pristup internetu velikom brzinom preko WiFi i mobilne mreže.

Na tržištu mobilnih telefona dominira nekoliko proizvođača (Apple, HTC, Nokia, RIM i dr.). Neki proizvođači sami razvijaju operativne sisteme za svoje telefone, a neki prilagođavaju svoje telefone nekim od postojećih operativnih sistema.

Prema istraživanju agencije Gartner, dominantnu poziciju na tržištu sudeći po broju prodanih mobilnih telefona u 2011. godini zauzima Nokia sa 23,8%, a zatim Samsung sa 17,7% [12].

Tabela 1. Prodaja mobilnih telefona, po proizvođačima u 2011. godini [12]

Proizvođač	Tržište 2011 (%)	Tržište 2010 (%)
Nokia	23.8	28.9
Samsung	17.7	17.6
Apple	5.0	2.9
LG Electronics	4.9	7.1
ZTE	3.2	1.9
Research In Motion	2.9	3.1
HTC	2.4	1.5
Huawei	2.3	1.5
Motorola	2.3	2.4
Sony Ericsson	1.8	2.6
Ostali	33.7	30.4
Ukupno	100	100

Najpopularniji operativni sistem na tržištu mobilnih telefona, u 2011. godini je operativni sistem Android, po istraživanju Canalys [13].

Tabela 2. Prodaja smart telefona širom sveta, podela po operativnim sistemima u 2011. godini [13]

Operativni sistem	Prodaja po komadu (milion)	Tržište 2011 (%)	Godišnji rast (%)
Android	237.7	48.8	244
iOS	93.1	19.1	96
Symbian	80.1	16.4	-29.1
BlackBerry	51.4	10.5	5.0
Bada	13.2	2.7	183.1
Windows Phone	6.8	1.4	-43.3
Ostali	5.4	1.1	14.4
Ukupno	487.7	100	62.7

4.1 Operativni sistemi pametnih telefona

Najčešće korišćeni operativni sistemi mobilnih telefona su Android (Google), Windows Mobile (Microsoft), Windows Phone (Microsoft), iOS (Apple), Symbian (Nokia) and BlackBerry OS (RIM).

4.1.1 Android

Android je nedavno razvijen operativni sistem za mobilne uređaje, kao što su pametni telefoni ili tablet računari. Razvijao ga je The Open Handset Alliance na čelu sa Google-om, a baziran je na Linux-ovom kernelu. Android OS je otvoren – open source (pod Apache licencom), a razvijaju ga programeri širom sveta.

Ovaj operativni sistem podržava mnoge hardverske komponente, npr. kameru, WiFi komunikacioni čip, mobilni komunikacioni čip, Bluetooth, ekran osetljiv na dodir, NFC (Near Field Communications) i dr.

Android SDK (Software Development Kit) obezbeđuje alate i API (Application Program Interface) neophodne za razvoj aplikacija na Android platformi u programskom jeziku Java. Android API sadrži mnoge funkcije i klase koje su u stanju da kontrolišu funkcionalnosti mobilnih uređaja.

U svetu postoji veliki broj programera koji rade na razvoju Android aplikacija (“apps”) koje mogu da nasleđuju osobine samih uređaja. Aplikacije se mogu postavljati i preuzimati sa “online” marketa Google Play.

Srž svakodnevnog korišćenja Androida je upotreba Google naloga preko Gmail-a, adresar, kalendar i mapa.

Do februara, 2012. godine postojalo je više od 450 000 aplikacija za Android, a po istraživanju Canalsys, Android je najprodavanija platforma ugrađena u pametne telefone u 2011. godini.

Nakon prvog pojavljivanja, Android ima brojne nadogradnje u kojima su dodate neke funkcionalnosti ili ispravljene greške (“bugs”). Svaka verzija Androida je imenovana nazivom deserta.

4.1.1.1 Istorija Android operativnog sistema

Tabela 3. Verzije Android OS sa osnovnim funkcionalnostima

Verzija Android OS	Osnovne funkcionalnosti
Verzija 1.0 23. septembar, 2008.	• download i update aplikacija sa Android Marketa • web browser • podrška za kameru • Gmail sinhronizacija • adresar, kalendar, Google maps, Google talk, Media player • WiFi i bluetooth podrška
1.5 Cupcake 30. april, 2009.	• virtuelna tastatura sa predikcijom • podrška za widget • video snimanje u MPEG-4 i 3GP formatu • animacije • autorotacija ekrana • mogućnost postavljanja videa na YouTube • mogućnost postavljanja slika na Picasa

1.6 Donut 15. septembar, 2009.	• dodata mogućnost za programere da dodaju svoje sadržaje u rezultate pretrage • bolja integracija kamere, galerije • podrška za WVGA rezolucije ekrana • olakšana pretraga aplikacija i mogućnost prikaza slika ekrana u Android Marketu
2.0/2.1 Eclair 26. oktobar, 2009.	• proširena sinhronizacija naloga, dodata mogućnost dodavanja više naloga za sinhronizaciju mailova i kontakata • Bluetooth 2.1 podrška • pretraga svih SMS i MMS poruka sa automatskim brisanjem • poboljšana brzina kucanja na soft tastaturi • podrška za više veličina ekrana i rezolucija sa boljim odnosom kontrasta • poboljšane Google Maps 3.1.2
2.2.x Froyo 20. maj, 2010.	• optimizovana brzina, memorija i performanse • integracija JavaScript Engine u web browser • podrška za Android Cloud to Device Messaging (C2DM) servis • USB tethering i Wi-Fi hotspot funkcionalnost • Opcija za onemogućavanje pristupa podacima preko mobilne mreže • brzo prebacivanje između višejezične tastature i rečnika • glasovno biranje i deljenje kontakata putem Bluetooth-a • podrška za Bluetooth-enabled car and desk docks • Adobe Flash podrška
2.3.x Gingerbread 06. decembar, 2010.	• unapređen korisnički interfejs sa povećanom brzinom i jednostavnošću • podrška za jako velike ekrane i rezolucije (WXGA i veće) • ugrađena podrška za SIP VoIP internet telefoniranje • podrška za Near Field Communication (NFC), koja omogućava korisniku da čita NFC tagove sa postera, stikera ili oglasa • audio, grafika i ulazna proširenja za programere igara • ugrađena podrška za senzore (kao što je žiroskop i barometar)
3.x Honeycomb 22. februar, 2010.	• nadogradnja specifična za tablet uređaje
4.x Ice Cream Sandwich 19. oktobar, 2011.	• poboljšana brzina i performanse • soft dugmići na korisničkom interfejsu, umesto hardverskih • poboljšana copy/paste funkcionalnost • Android Beam, lokacijska komunikacija koja omogućava deljenje kontakata, YouTube videa i drugih podataka • podrška za WebP format slika • hardverska akceleracija korisničkog interfejsa • Wi-Fi Direct

Android 4.0 (API nivo 14) uvodi podršku za Bluetooth Health Device Profile (HDP). Ovo omogućava kreiranje aplikacija koje koriste Bluetooth za povezivanje sa zdravstvenim uređajima koji podržavaju Bluetooth, kao što su merači otkucaja srca, merači stanja krvi, toplomeri i vage.

Već sada postoji više od 40 miliona kućnih i profesionalnih zdravstvenih uređaja sa

bluetooth-om na tržištu od vodećih proizvođača kao što su 3M, A&D, Nonin i Omron. Analitičko-istraživačka firma IMS procenjuje da će se do 2013. godine prodavati jedna milijarda niskoenergetskih bluetooth uređaja - do sada najbrža stopa usvajanja bilo koje bežične tehnologije.

Rastuća svest o dobrobiti zdravog života će stvoriti procvat tržišta uređaja koji omogućavaju kontrolisanje zdravlja u vlastitom domu.

Bluetooth tehnologija je idealna bežična tehnologija za povezane zdravstvene i fitness uređaje zato što je jednostavna, sigurna i sveprisutna [15].

4.1.2 iPhone

iPhone ili iOS je operativni sistem mobilnih telefona koga je razvio Apple. Prvi put je izbačen u upotrebu 2007. godine za iPhone i iPod Touch, a kasnije je nadograđen kako bi bio podržan i od strane drugih uređaja kao što su iPad i Apple TV. Za razliku od Windows i Android operativnog sistema, Apple ne licencira iOS za instalacije na drugim uređajima (koji nisu Apple).

iOS je izveden iz Mac operativnog sistema. iOS se sastoji iz četiri apstraktna sloja: Core OS sloj, Core Services sloj, Media sloj i Cocoa Touch sloj. Trenutna verzija iOS 5.1 koristi otprilike 770 mb memorijskog prostora u uređaju.

U martu, 2012. godine, App Store je sadržavala preko 550 000 iOS aplikacija, koje su sve zajedno bile preuzete više od 25 milijardi puta.

Korisnički interfejs iOS operativnog sistema je baziran na konceptu direktne interakcije, korišćenjem dodira. Korisničke kontrole se sastoje iz "slajdera", "switch" dugmadi i običnih dugmića. Interakcija sa OS uključuje dodire kao što su "swipe", "tap", "pinch" i obrnuti "pinch", a svaki od njih ima specifične definicije u okviru iOS.

Interni akcelerometri se koriste u nekim aplikacijama i reaguju na pomeranje uređaja (najčešća reakcija je "undo" operacija) ili rotaciju u tri dimenzije (prelazak iz "portret" moda u "landscape").

Aplikacije se pišu i kompajliraju specijalno za iOS i ARM arhitekturu (32-bitna arhitektura). Safari "web browser" podržava web aplikacije.

U oktobru 2007. godine S. Jobs je objavio da će SDK biti dostupan i trećim licima, pa su od marta, 2008. godine programeri bili u mogućnosti da razvijaju aplikacije za iPhone i iPod touch i testiraju ih na iPhone emulatoru. Od aprila 2012. godine uvedena je taksa za programere za iOS i OS X (iPhone Developer i Mac Developer) od 99\$ godišnje. Ovo omogućava programerima da postave svoje aplikacije na App store i ostvare određenu zaradu od prodatih aplikacija. Postoji i mogućnost postavljanja besplatne aplikacije i u tom slučaju se plaća samo članarina.

Xcode je programsko okruženje za iOS SDK. iPhone aplikacije su napisane u Objective-C jeziku [18].

4.1.3 Symbian operativni sistem

Symbian OS je originalno razvijen od strane Symbian Ltd.

Symbian operativni sistem i platforma su dizajnirani za pametne telefone. Symbian platform je naslednik Symbian OS kojem je poboljšan korisnički interfejs.

Svi modeli OS imaju nekoliko slojeva:

- UI Framework sloj
- Application Services sloj (Java ME)
- OS Services sloj
- Base Services sloj
- Kernel Services & Hardware Interface sloj

Od 2010. godine Symbian koristi C++ jezik kao osnovni SDK. Alternative za razvoj aplikacija su Python, Adobe Flash Lite, Ruby ili Java ME.

Java ME aplikacije za Symbian OS su razvijane koristeći standardne tehnike i alate kao Sun Java Wireless Toolkit (J2ME Wireless Toolkit). Takve aplikacije su pakovane u tzv. JAR fajlove (uz dodatak JAD opisnog fajla). I CLDC (Connected Limited Device Configuration) i CDC (Connected Device Configuration) aplikacije mogu da se kreiraju sa NetBeans okruženjem.

Symbian 3 i ranije verzije imaju ugrađeni WebKit browser. Raniji Symbian uređaji imaju Opera Mobile kao osnovni web browser.

Trenutna verzija Symbian OS ima podršku za 48 jezika.

Aplikacije razvijene za Symbian telefone mogu da budu distribuirane do korisničkih telefona preko tačne putanje. Pakuju se u SIS fajlove i mogu da budu instalirane "over-the-air", preko spojenog računara, Bluetooth-a ili preko memorijske kartice.

Da bi obezbedile neke specifične funkcionalnosti aplikacije treba da budu potpisane Symbian potpisom [21].

4.1.4 Windows Mobile

Windows Mobile je operativni system kojeg je razvio Microsoft za pametne telefone i džepne računare. Osnovu čini Windows CE kernel. Prvo pojavljivanje ovog OS je bilo bazirano na paketu osnovnih aplikacija razvijenih sa Microsoft Windows API sa funkcijama sličnim kao na desktop windows računarima.

Prvi Windows Mobile telefoni su se prodavali sa olovkama koje su korišćene za unos komadi preko ekrana osetljivog na dodir.

U februaru 2010. godine Microsoft je najavio novu platformu za mobilne telefone – Windows Phone. Telefoni koji koriste Windows Mobile ne mogu da koriste softver pisan za Windows Phone.

Većina verzija Windows Mobile operativnog sistema ima standardne osobine, kao multitasking i mogućnost navigacije kroz fajl sistem uz podršku mnogih ekstenzija.

Kao i kod desktop računara OS dolazi sa dosta ugrađenih aplikacija koje omogućavaju izvršenje osnovnih zadataka na telefonu.

Internet Explorer Mobile je glavni internet browser, a Windows Media Player se koristi za muziku i video. Microsoft Office Mobile je mobilna verzija Office aplikacija.

Postoje tri osnovne verzije Windows Mobile OS za različite tipove uređaja:

- Windows Mobile Professional za pametne telefone sa ekranom osetljivim na dodir
- Windows Mobile Standard za mobilne telefone bez ekrana osetljivog na dodir
- Windows Mobile Classic za PDA uređaje ili džepne računare

Softver za Windows Mobile operativni sistem mogu da razvijaju treća lica. Programeri imaju nekoliko opcija za razvoj mobilnih aplikacija: mogu da pišu nativni kod sa Visual C++ ili souce kod koji radi pod .NET Compact Framework-om.

.NET Compact Framework je podskup .NET framework-a tako da je omogućeno deljenje softverskih resursa sa desktop računarima, serverima i web serverima koji imaju instaliran .NET framework.

Kao pomoć programerima Microsoft je obezbedio SDK koji radi sa Visual Studio okruženjem za razvoj softvera. SDK uključuje i emulatore za testiranje aplikacija [19].

4.1.5 Windows phone

Windows Phone je operativni sistem za mobilne telefone kojeg je razvio Microsoft i sledbenik je Windows Mobile OS.

U upotrebi je od sredine 2010. godine. Ovim OS je obezbeđen novi korisnički interfejs, baziran na Windows Phone dizajnu zvanim Metro. Početni ekran se sastoji iz "Live tiles" kockica raspoređenih na ekranu koje su povezane sa aplikacijama ili određenim funkcijama.

Nekoliko funkcija je organizovano u tzv. "hubs" koji čine kombinaciju lokalnih i online sadržaja.

Windows Phone koristi "multi-touch" tehnologiju.

Osnovni web browser je Internet Explorer Mobile koji je prostija verzija Internet Explorer-a 9.

Aplikacije, muzika, video i ostali sadžaj može da se postavi na Windows Phone Marketplace, a dostupan je preko softverskog alata Zune. Marketplace održava Microsoft, a dostupan je u 54 zemlje.

Aplikacije koje razvijaju programeri treba da su bazirane na XNA (set alata za upravljanje radnim okruženjem) ili WP7 verziji Silverlight (aplikacioni framework) [20].

4.1.6 BlackBerry

BlackBerry je linija mobilnih email i pametnih uređaja koje je razvila i dizajnirala kanadska kompanija RIM (Research In Motion) 1999. godine.

BlackBerry uređaji su pametni telefoni, dizajnirani tako da služe kao PDA uređaji, prenosivi media player-i, browser-i, uređaji za igranje i dr. Kada su se pojavili bili su poznati po tome što su bili u mogućnosti da primaju i šalju mailove i poruke preko visoko zaštićenih i enkriptovanih kanala.

Operativni sistem u BlackBerry uređajima je odgovarajuće multitasking okruženje koje je razvio RIM. OS 5.0 obezbeđuje podršku za Java MIDP 2.0 i podršku za wireless sinhronizaciju sa emailom, kalendarom, taskovima, kontaktima Microsoft Exchange Server-a i emailom Lotus Domino-a [22].

4.2 Zajedničke karakteristike mobilnih uređaja

Mobilni telefoni se danas koriste u različite svrhe, privatne ili poslovne i na različitim mestima, a prema [10] postoji nekoliko jedinstvenih osobina mobilnih uređaja kao najmodernijeg vida komunikacije:

1. mobilni uređaj je ličan
2. uvek se nosi sa sobom
3. uvek je uključen
4. poseduje ugrađene kanale plaćanja (pre-paid ili post-paid)
5. odličan je za sprovođenje različitih stvaralačkih i zanimljivih ideja
6. omogućene su povratne informacije, a u skladu sa njima i način daljeg sprovođenja programa ili ideja
7. omogućava obraćanje pojedincima ili grupama u pravo vreme i na pravom mestu

Sve ovo čini mobilni uređaj najadekvatnijim kanalom za usmeravanje zdravstvenih informacija i pružanje zdravstvenih usluga za razliku od drugih vidova komunikacije (TV, radio, novine, e-mailovi i sl.)

5 Dostupne tehnologije i alati za implementaciju mHealth sistema

5.1 XML (Extensible Markup Language)

XML ili proširivi markerski jezik je najnoviji jezik za komunikaciju na Internetu, tehnologija koja se svakim danom sve više razvija i koja se koristi prevashodno za upravljanje, prikazivanje i organizovanje podataka [27].

5.1.1 Pojam XML-a

XML predstavlja tehnologiju koja je orijentisana ka opisivanju i struktuiranju podataka. Zahvaljujući svojoj visokoj fleksibilnosti, XML može da se koristi kao osnova za definisanje jezika za razmenu podataka, i to prevashodno u cilju komunikacije preko Interneta. Jedan takav jezik bi rad sa podacima u okviru neke aplikacije učinio veoma jednostavnim, ali bi isto tako olakšao i deljenje takvih informacija sa drugim korisnicima [1]. XML dokumenti se mogu kreirati, raščlanjivati i prenositi između aplikacija koje se izvršavaju na odvojenim platformama. Drugim rečima, svaki server može opsluživati svakog klijenta koji može da kreira i raščlani XML [28].

5.1.2 Karakteristike XML-a

Neke od karakteristika XML-a su:

- Standardizovani format fajla

XML ima jednostavnu sintaksu, čitljiv je za ljude. Format XML-a je standardizovan od strane W3C. Uprkos relativnoj novosti XML, raščlanjivači su već dostupni za svaki glavni programski i skriptni jezik, tako da se može čitati bez obzira na platformu ili jezik. XML raščlanjivači proveravaju da li su dokumenti „dobro oblikovani“ i mogu podići greške kada čitaju dokument koji sadrži greške.

- Samoopisivanje

Za razliku od jednostavnih formata tekst fajlova sa odvajanjima ili fiksnom širinom polja, XML fajlovi koriste simboliku oznaka za opisivanje tipa podataka uključenih u svako polje. Dobro odabrana imena oznaka poboljšavaju čitljivost i u programiranju koda i pri pregledanju čistih XML dokumenata.

- Hijerarhija

Za razliku od formata tekst fajlova, običnog fajla sa odvajanjem ili fajla sa fiksnom širinom polja, XML dokumenti mogu predstavljati proizvoljan broj hijerarhijskih relacija.

Mogućnost modeliranja relacija takođe znači da se može koristiti ime oznake oznake na odgovarajući način, čak i ako to duplira ime oznake koja se pojavljuje negde na

drugom mestu u dokumentu.

- Mogućnost prihvatanja i struktuiranih i nestruktuiranih podataka

Relacione baze podataka već obezbeđuju način za skladištenje hijerarhijskih struktuiranih podataka, ali nisu idealne za skladištenje nestruktuiranih podataka jer najbolje rade kada se može jasno definisati relacija između različitih stavki i grupa stavki. XML dokumenti mogu modelirati relacione šeme vrlo lako, tako da su eminentno odgovarajući za hijerarhijski struktuirane podatke, ali pošto nema zahteva za modeliranje hijerarhijskih relacija, XML dokumenti rade podjednako dobro i za nestruktuirane podatke.

- Podrška upita

Koristeći standardizovanu gramatiku nazvanu Xpath, može da se konstruiše upit XML dokumentu ili da se pronađu i broje XML elementi, atributi i vrednosti. Za razliku od toga nema standardizovanog načina za pronalaženje stavke podataka u tekst fajlu. Baze podataka podržavaju upite i optimizovane su za velike setove podataka, dok XML dokumenti nisu. Koristeći kombinaciju ova dva pristupa, mogu da se modeliraju relacije iz veće baze podataka u okviru mnogo manjeg podskupa podataka u XML. To je ideja u pozadini ADO.NET setova podataka.

- Prolaz kroz firewall

XML je sastavljen od čistog teksta; zbog toga, on može lako prolaziti kroz firewall i omogućiti kompanijama da izlože interni kod eksternim klijentima bez uticaja na bezbednost.

- Podrška šema i provera validnosti

W3C je nedavno izdao preporuku XML šema koja koristi drugi XML dokument za opisivanje XML dokumenta. Kako su XSD šeme, XML dokumenti mogu se programski čitati, što čini komunikacije između računara mogućim čak i kada računari koji komuniciraju ne znaju unapred kompletan protokol. Koristeći raščlanjivač za proveru validnosti može da se proveru sadržaj dokumenta prema šemi, obezbeđujući da dokument sadrži sve elemente i attribute koje bi trebalo da sadrži. Pojava šeme, XML-a, robusnih raščlanjivača i naravno sveprisutnih HTTP veza i Interneta čini Web servise mogućim [28].

5.1.3 Upotreba XML-a

XML je nezavisna platforma i jezik. Kada je potrebno uspostaviti komunikaciju između raznih računarskih programa, XML predstavlja dobar izbor formata za razmenu. XML koriste i mrežno orijentisane aplikacije u cilju smanjenja opterećenosti servera (informacije se čuvaju na klijentu, a zatim se prosleđuju ostalim klijentima u vidu XML dokumenta). Na nekim Web lokacijama se celokupan sadržaj obrađuje u XML-u. Taj XML se pomoću XSLT-a pretvara u HTML. XML se koristi i kao protokol za daljinsko povezivanje procedura (Remote Procedure Calls, RPC). RPC je protokol koji omogućava objektima sa jednog računara da, u cilju obavljanja određenih aktivnosti, pozivaju objekte sa nekog drugog računara, vršeći na taj način

raspodeljenu obradu. I konačno, javni interfejs Web servisa je definisan i opisan u XML-u [28].

5.2 JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) je jednostavan format za razmenu podataka. Za ljude je čitljiv i razumljiv, a za mašine lagan za parsiranje i generisanje.

Bazira se na osnovama JavaScript programskog jezika.

JSON je tekstualni format, u potpunosti jezički nezavisan, sa konvencijama koje su poznate programerima C familije jezika. Ove osobine čine JSON idealnim jezikom za razmenu podataka.

JSON se bazira na dve strukture:

- *Kolekciji ime/vrednost para.* U mnogim jezicima, ovo se realizuje kao objekat, rekord, struktura, rečnik, hash tabela, niz.
- *Uređena lista vrednosti.* U mnogim jezicima, ovo se realizuje kao niz, vektor, lista ili sekvenca.

Ovo su univerzalne strukture podataka. Praktično, svi moderni programski jezici podržavaju JSON u jednom ili drugom obliku.

Objekat je neuređena lista ime/vrednost para. Objekat počinje otvorenom zagradom { i završava zatvorenom }. Svako ime prati dvotačka, a ime/vrednost je odvojeno zarezom [29].

Sličnosti JSON sa XML-om:

- JSON je običan tekst
- JSON je samoopisujući (čitljiv za ljude)
- JSON je hijerarhijski (vrednosti unutar vrednosti)
- JSON parsira JavaScript
- JSON podaci mogu da se prenose preko AJAX-a

Za razliku od XML-a:

- Nema završni tag
- Kraći je
- Brže se čita i piše
- Može da se parsira koristeći ugrađeni JavaScript eval()
- Koristi nizove
- Nema rezervisane reči

Ekstenzija JSON fajlova je .json, a MIME (Multipurpose Internet Mail Extensions) tip za JSON text je "application/json".

5.3 Web servisi

Web servisi su programske komponente koje omogućavaju pravljenje skalabilnih, slabo povezanih i platformski nezavisnih aplikacija. Infrastruktura web servisa se bazira na određenim standardima, kao što su Simple Object Access Protocol (SOAP), opisni jezik web servisa (WSDL) i Universal Description, Discovery and Integration (UDDI) [32].

Web servis je komponenta koja implementira programsku logiku i omogućava funkcionalnost za različite aplikacije [30].

Važna karakteristika modela na kojem je zasnovan Web servis je da klijent ne mora da poznaje jezik u kojem su Web servisi implementirani. Klijent samo mora da zna lokaciju Web servisa i metode koje može da pozove za taj servis. Klijentske aplikacije mogu da pozovu metodu servisa i da proslede potrebne informacije kao argumente u pozivu metode. Podaci koji su vezani za poziv metode i za argumente šalju se Web servisu u XML formatu korišćenjem protokola SOAP preko HTTP transportnog mehanizma [33]. XML je srce Web servisa i obezbeđuje velike prednosti nad postojećim metodama za pozivanje koda koji se izvršava na udaljenim serverima.

5.3.1 Pojam Web servisa

Web servis (Web service) je softverski sistem identifikovan sa URI (Unified Resource Identifier)² čiji je javni interfejs definisan i opisan u XML-u. Njegovu definiciju mogu da pronađu drugi softverski sistemi i kad ga pronađu, mogu sa njim da stupe u interakciju na predefinisani način, korišćenjem XML poruka prenesenih preko Internet protokola.

Web servisi predstavljaju standardizovan način povezivanja Web aplikacija korišćenjem skupa otvorenih standarda (XML, SOAP, WSDL, UDDI) i Internet protokola kao osnove [30].

XML služi za formatiranje podataka, SOAP za formatiranje poruka kojima se prenose podaci, WSDL za opis Web servisa i UDDI za popis dostupnih Web servisa.

Prvobitno, Web servisi su zamišljeni kao sredstvo za bussines-to-bussines (B2B) i bussines-to-client (B2C) komuniciranje, koje omogućavaju razmenu podataka između organizacija bez potrebe za međusobnim poznavanjem karakteristika informacionog sistema sa kojim se komunicira.

Danas, Web servisi predstavljaju osnovu za izgradnju interoperabilnih distribuiranih i mobilnih aplikacija.

² Adresna shema na Web-u koja omogućava da se neki resurs na njemu jedinstveno identifikuje. Različite vrste Web resursa zahtevaju različite vrste URI-ja. URL (Uniform Resource Locator) definiše adresu nekog resursa, dok URN (Uniform Resource Name) definiše konstantno ime resursa, nezavisno od lokacije.

Za razliku od tradicionalnog klijent/server modela (npr. Web server/Web stranica) Web servisi nemaju grafički korisnički interfejs, već se kroz programski interfejs (API) razmenjuju podaci i poslovna logika.

U slučaju Web servisa međusobno komuniciraju aplikacije, a ne korisnici.

Korisnici Web servisa mogu biti drugi Web servisi, Web/Win ili mobilne aplikacije (koje omogućavaju interakciju sa krajnjim korisnicima). Ponegde, Web servisi se nazivaju i aplikacioni servisi (application services) [30].

5.3.2 Arhitektura Web servisa

Kako bi se operacije oglašavanja, pronalaženja i povezivanja izvršile na interoperabilan način mora postojati stek Web servisa koji utvrđuje standarde na svakom nivou. Na slici 4. je prikazan konceptualni stek Web servisa. Viši slojevi izgrađeni su na mogućnostima koje nude niži slojevi.

WSFL Web Services Flow Language	Tok servisa
UDDI Universal Description, Discovery and Integration	Otkrivanje servisa
UDDI Universal Description, Discovery and Integration	Objavljivanje servisa
WSDL Web Services Description Language	Opis servisa
SOAP Simple Object Access Protocol	Razmena XML poruka
Transportni protokoli HTTP/FTP/SMTP/JMS/IIOP ...	Mreža

Slika 4. Stek Web servisa [30]

U osnovi steka Web servisa je mreža. Razlog ovome je što Web servisi moraju biti dostupni na mreži kako bi ih tražioci servisa mogli pozvati. Javni Web servisi na Internetu koriste uobičajene mrežne protokole. S obzirom da je opšte prihvaćen, HTTP je 'de facto' standardni mrežni protokol za Web servise na Internetu. Pored njega mogu biti podržani i drugi Internet protokoli kao što su SMTP (Simple Mail Transfer Protocol) ili FTP (File Transfer Protocol).

Sledeći sloj, razmena poruka baziranih na XML-u, predstavlja korišćenje XML-a kao osnove protokola za razmenu poruka i za njega je izabran SOAP protokol, i to iz sledećih razloga:

- to je standardizovani mehanizam za komunikaciju poruka zasnovanih na dokumentima i poziv udaljenih procedura (remote procedure call) korišćenjem XML-a;
- jednostavan je i u osnovi je HTTP POST sa XML envelope kao korisnom

nosivošću;

- koristi se umesto jednostavnog HTTP POST-a, zato što definiše standardni mehanizam za proširenja poruke korišćenjem SOAP header-a i standardno kodiranje operacija ili funkcija;
- SOAP poruke podržavaju operacije objavljivanja, pronalaženja i povezivanja u arhitekturi Web servisa.

I sloj za opis servisa se bazira na XML-u; Web Services Description Language (WSDL) je standard za opis servisa koji se baziraju na XML-u. Ovaj opis servisa predstavlja minimum koji je neophodan da bi se podržala interoperabilnost Web servisa. WSDL definiše interfejs i mehanizme interakcije servisa. WSDL dokument može biti dopunjen i drugim dokumentima kojima se daje opis servisa, kako bi se dali i neki drugi njegovi aspekti.

	WSDL Web Services Description Language	Opis servisa	
	SOAP Simple Object Access Protocol	Razmena XML poruka	
	HTTP Hyper Text Transfer Protocol	Mreža	

Slika 5. Stek interoperabilnosti Web servisa [30]

Pošto je Web servis definisan da bude dostupan (na mreži) preko SOAP-a, i predstavljen preko opisa servisa, ova tri sloja steka Web servisa su neophodna da bi se obezbedio ili koristio bilo koji Web servis. Najjednostavniji stek bi se sastojao od HTTP-a za mrežni sloj, SOAP protokola za sloj razmene XML poruka i WSDL-a za sloj opisa servisa. Ovo je osnovni, minimalni stek koji svi Web servisi moraju da podržavaju da bi se ispunio uslov interoperabilnosti i on je prikazan u nastavku.

Dok donja tri sloja steka Web servisa (sa slike 4.) određuju tehnologije u cilju postizanja saglasnosti i interoperabilnosti, dva sloja - objavljivanje i otkrivanje servisa, mogu se implementirati na različite načine.

S obzirom da Web servis ne može biti pronađen, ukoliko prethodno nije objavljen, može se reći da pronalaženje servisa zavisi od objavljivanja servisa. Svaki mehanizam koji omogućava tražiocu da pristupi opisu servisa i učini ga dostupnim aplikaciji u vreme izvršavanja, kvalifikuje se kao pronalaženje servisa. Najjednostavniji primer statičkog pronalaženja je onaj kod koga tražilac servisa prihvata WSDL dokument iz lokalne datoteke. Ovo je obično WSDL dokument, dobijen direktnim objavljivanjem ili kao rezultat operacije pronalaženja. Isto tako

pružalac servisa može da objavi WSDL dokument kojim opisuje servis u lokalnom WSDL registru, privatnom UDDI registru ili čvoru UDDI operatora.

Implementacija Web servisa je softverska komponenta, pa je i prirodno da Web servisi mogu nastati kao rezultat komponovanja drugih Web servisa. Ovo komponovanje može imati više primena.

Web servisi organizacija mogu međusobno sarađivati kako bi proizveli jedinstveni Web servis interfejs za javnost. Isto tako Web servisi različitih organizacija mogu sarađivati kako bi se proizvele machine-to-machine, bussines-to-bussines transakcije. Takođe, workflow manager može da poziva svaki Web servis kao učesnika u poslovnom procesu.

Najviši sloj, tok servisa, opisuje kako se izvršavaju service-to-service komunikacije, saradnje i tokovi. WSFL se koristi da bi se opisale ove interakcije. Donji slojevi koji predstavljaju osnovu Web servis steka, su razvijeniji i standardizovaniji od viših slojeva steka. Razvijanje i usvajanje Web servisa vodiće standardizaciji i viših slojeva steka [30].

5.4 SOAP (Simple Object Access Protocol)

Iako XML reducira proces udaljenih komunikacija na jednostavne XML – formatirane tekstualne fajlove, da bi programi bili interoperativni, potrebno je raspolagati standardom koji specificira format zahteva i odgovora. Jedan od standarda je Simple Object Access Protocol (SOAP).

SOAP je komunikacioni protokol, baziran na XML-u, za razmenu informacija između računara bez obzira na njihov operativni sistem, programsko okruženje ili objektni model.

U okviru specifikacije, SOAP je definisan kao jednostavan protokol za razmenu informacija između uređaja ravnopravnih komunikacionih mogućnosti (peers) u decentralizovanom i distribuiranom okruženju. Ovo okruženje se obično odnosi na mreže kao što je Internet, ali to nije uvek slučaj. SOAP se jednostavno može iskoristiti i za komunikaciju između računara u lokalnoj računarskoj mreži (Local Area Network, LAN). SOAP ne definiše programski model ili semantiku određene implementacije. On definiše jednostavan mehanizam za razmenu poruka, tako što obezbeđuje model za modularno pakovanje i način kodiranja podataka unutar poruka koji se bazira na XML šemi [30].

5.5 WSDL – Web Service Description Language

SOAP obezbeđuje standardan način za transport poruka koje koriste Web servisi, ali ne obezbeđuje način na koji bi se opisao format ovih poruka za dati Web servis.

Ubrzo posle pojavljivanja SOAP specifikacije, pojavile su se i prve implementacije Web servisa baziranih na SOAP-u. Međutim nastao je problem. Jedinu način da se korisnicima Web servisa objasni kako da ga pozivaju, jeste da se objavi primer SOAP dokumenta koji ilustruje ovaj poziv. Problem kod ovog pristupa je u tome što ne obezbeđuje dovoljno informacija o samom postupku pozivanja Web servisa.

Na strani koja vrši prijem (server), pisanje kôda kojim se parsira određena SOAP poruka i njeno transformisanje u poziv binarne funkcije ne predstavlja problem. Međutim, na strani koja vrši poziv (klijent) pojavljuje se problem u vezi sa kreiranjem SOAP poruke (poziva) zbog nepostojanja "header datoteke" kojim se opisuje kreiranje SOAP poruke.

Kao rešenje ovih problema stvoren je WSDL. Njime se definišu i opisuju Web servisi, baš kao što header datoteke definišu i opisuju tradicionalne binarne biblioteke.

Kako bi se ostvarila vizija o Web servisima u smislu miliona nezavisnih komponenata dostupnih preko Interneta i upotrebljivih na bilo kojoj platformi i svim razvojnim jezicima, oni moraju biti samoopisujući. Ovo je omogućeno WSDL-om, kojim se opisuju njihovi interfejsi i povezivanja. Ukoliko bi svi Web servisi objavljivali njihove interfejse korišćenjem WSDL-a, klijenti Web servisa bi mogli da ih pozivaju programski, bez potrebe da znaju na koji način je izvršena implementacija Web servisa, niti informacije o platformi ili operativnom sistemu na kome se servis izvršava.

WSDL je izgrađen na otvorenim standardima kao što su XML, XSD, SOAP i MIME kako bi se omogućilo da Web servisi budu nezavisni od platforme, implementacije i uređaja. WSDL obezbeđuje gramatiku za opisivanje servisa kao skupa krajnjih tačaka koje razmenjuju poruke [30].

5.6 REST (Representational state transfer)

REST definiše set arhitektonskih principa na kojima se zasniva razvoj Web servisa, a koji se prvenstveno baziraju na sistemskim resursima, uključujući adresiranje i transport preko HTTP do velikog broja klijenata napisanih u različitim jezicima.

Po broju Web servisa koji ga koriste, REST je u poslednjih nekoliko godina dominantni dizajn model za razvoj Web servisa.

Ustvari, REST je imao veliki uticaj na Web da je umalo zamenio SOAP i WSDL dizajn jer je jednostavniji za korišćenje.

REST nije privukao mnogo pažnje kada je prvi put predstavljen 2000. godine na Univerzitetu u Kaliforniji, u doktorskoj disertaciji Roy Fieldinga [31]. Godinama nakon predstavljanja, pojavljuju se značajni okviri za REST, a i dalje se razvijaju.

Konkretna implementacija REST Web servisa prati nekoliko osnovnih principa:

- Eksplicitno korišćenje HTTP metoda
- Stateless (bez stanja)

- URI sa strukturom direktorijuma
- Prenos XML, JavaScript Object Notation - JSON ili oba.

5.6.1 Eksplicitno korišćenje HTTP metoda

Jedna od ključnih karakteristika RESTful Web servisa jeste eksplicitno korišćenje HTTP metoda na način koji prati protokol definisan sa RFC 2616.

Osnovni REST dizajnerski princip uspostavlja jedan-na-jedan mapiranje između kreiranja, čitanja, ažuriranja i brisanja (create, read, update, and delete) – CRUD operacija i HTTP metoda.

Prema tome:

- Za kreiranje resursa na serveru koristi se POST
- Za dohvatanje resursa koristi se GET
- Za promenu stanja resursa, odnosno ažuriranje koristi se PUT
- Za brisanje resursa koristi se DELETE

5.6.2 Stateless (bez stanja)

REST Web servis aplikacija ili klijent sadrži u okviru HTTP zaglavlja i tela sve parametre, kontekst i podatke potrebne serverskoj komponenti koja će da kreira odziv.

“Stateless” ili karakteristika bez stanja u ovom smislu, poboljšava performanse Web servisa i pojednostavljuje dizajn i implementaciju serverskih komponenti jer upravo zbog odsustva stanja na serveru nije potrebno sinhronizovati podatke o sesiji sa eksternom aplikacijom.

5.6.3 URI sa strukturom direktorijuma

REST Web service URI treba da bude intuitivan. URI treba posmatrati kao vrstu samoopisujućeg interfejsa, koji zahteva malo ili nimalo dodatnih objašnjenja ili referenci za programera koji treba da razume šta je poenta Web servisa i da mu prosledi potrebne resurse. Struktura URI-ja treba da je jednoznačna, predvidljiva i lako razumljiva. Jedan od načina za postizanje ovih zahteva jeste organizacija URI-ja po direktorijumskoj strukturi. Takav tip URI-ja je hijerarhijski, sa početkom od jedne tačke i grananjem ka podputanjama. Postoji i nekoliko karakteristika URI strukture za REST Web servise:

- Sakrivaju ekstenziju fajlova (.jsp, .php, .asp), tako da je moguće isti URI usmeriti na sasvim drugi resurs bez menjanja njegove strukture
- Mogu da se ispisuju malim slovima
- Razmaci su zamenjeni sa crticama ili donjim crtama
- Potrebno je izbegavati “query strings”

5.6.4 Prenos XML, JSON ili oba

RESTful Web servis dizajn treba da sadrži podatak o tome koji tip podataka će razmenjivati klijent i server u HTTP telu zahteva, odnosno odziva.

Da bi klijentska aplikacija bila u mogućnosti da zahteva specifični tip sadržaja koji joj najviše odgovara, servis mora da ima ugrađeno HTTP Accept zaglavlje sa vrednostima MIME tipa. Najčešće korišćeni MIME tipovi su: JSON - application/json; XML - application/xml; XHTML - application/xhtml+xml.

To omogućava servisu da se koristi u raznim klijentskim aplikacijama pisanim različitim programskim jezicima sa izvršenjem na različitim platformama. Korišćenje MIME tipova i HTTP Accept zaglavlja je mehanizam poznat kao "content negotiation" – pregovaranje o sadržaju, koji klijentima omogućava da odaberu odgovarajući format podataka.

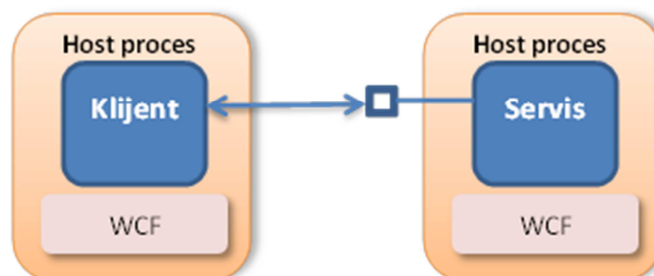
5.7 WCF - Windows Communication Foundation

WCF je programska platforma i izvršno okruženje za razvoj i konfigurisanje mrežno-distribuiranih servisa. To je poslednja servisno orijentisana tehnologija. Njena osnovna karakteristika je interoperabilnost. WCF čini jedinstveni programski model u okviru .NET framework-a, a čine ga kombinacije karakteristika Web servisa, MSMQ i COM+ [34].

5.7.1 Rešavanje problema: šta WCF obebeđuje

WCF je primarno implementiran kao set klasa nad .NET Framework's Common Language Runtime (CLR).

Ovo omogućava .NET programerima da razvijaju servisno orijentisane aplikacije na uobičajen način.



Slika 6. WCF-based klijenti i servisi mogu da se izvršavaju u bilo kojem Windows procesu

WCF omogućava razvoj klijenata koji pristupaju servisima. I klijent i servis mogu da

se pokrenu u gotovo svakom Windows procesu – WCF ne definiše neophodan host. Gde god da se izvršavaju, klijent i servis mogu da stupe u interakciju preko SOAP-a, WCF specifičnog binarnog protokola ili na drugi način.

WCF rešava niz problema u vezi sa komunikacijom između aplikacija. Kao najvažniji aspekti WCF-a, izdvajaju se tri stvari:

- Sjedinjavanje originalnih .NET Framework komunikacionih tehnologija
- Interoperabilnost sa aplikacijama razvijenim u drugim tehnologijama
- Eksplicitna podrška za servisno - orijentisani razvoj [35].

5.7.1.1 Sjedinjavanje originalnih .NET Framework komunikacionih tehnologija

Da bi se implementirala prosečna servisno orijentisana aplikacija u ranijim verzijama .NET Framework-a, bilo bi potrebno koristiti više od jedne komunikacione tehnologije, a često i sve njih kako bi se zadovoljili svi zahtevi, kao npr.:

- ASMX, koji se naziva i ASP.NET Web servis, kao opcija za komunikaciju sa Java EE aplikacijama
- .NET Remoting kao prirodno rešenje za komunikaciju sa call centrima. Remoting je dizajniran za .NET-to-.NET komunikaciju
- Enterprise Services za upravljanje objektima i definisanje distribuiranih transakcija
- Web Services Enhancements (WSE), zajedno sa ASMX za komunikaciju sa Java EE aplikacijama
- System.Messaging koji obezbeđuje programski interfejs za Microsoft Message Queuing (MSMQ), može da komunicira sa Windows-based aplikacijama koji nisu permanentno dostupne, itd.

Ovakva aplikacija, iako bi zadovoljila tražene zahteve, bila bi kompleksna za održavanje i razvoj. Da bi se to izbeglo moguće je upotrebiti bolje rešenje, kao na sledećoj slici:

	ASMX	.NET Remoting	Enterprise servisi	WSE	System. Messaging	System. Net	WCF
Interoperabilni Web servisi	X						X
Binarna .NET-.NET komunikacija		X					X
Distribuirane transakcije i sl.			X				X
Podrška za WS-* specifikaciju				X			X
Queued Messaging					X		X
RESTful komunikacija						X	X

Slika 7. Umesto korišćenja različitih tehnologija za različite vrste komunikacija, WCF obezbeđuje jedinstveno rešenje

Rezultat ovakve unifikacije je mnogo veća funkcionalnost i, pre svega, manja kompleksnost.

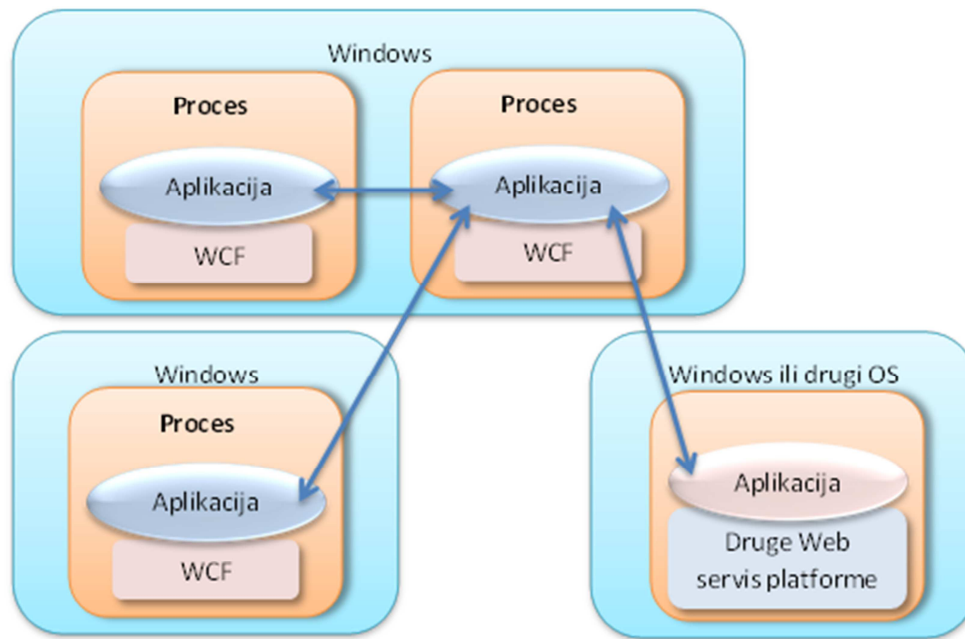
WCF omogućava aplikacijama da zadovolje sve komunikacione zahteve, a sama implementacija je olakšana, za razliku od ranijeg perioda gde se često koristila cela kolekcija tehnologija.

Obzirom da Microsoft i dalje podržava sve ranije tehnologije, nove aplikacije u kojima se koriste iste tehnologije, mogu da se “builduju” pod WCF [35].

5.7.1.2 Interoperabilnost sa aplikacijama razvijenim u drugim tehnologijama

Obzirom da je većina organizacija heterogena, WCF je dizajniran kao interoperabilan sa platformama drugih proizvođača, a ima i interoperabilnost sa drugim Microsoft tehnologijama koje su prethodile WCF.

Na slici je prikazana WCF aplikacija u komunikaciji sa drugim softverom u različitim okruženjima: u različitim procesima na istoj Windows mašini, WCF aplikacije koje se izvršavaju na drugoj Windows mašini, aplikacije razvijene u drugim tehnologijama, kao npr. Java EE aplikativni server koji podržavaju standardne Web servise i mogu da se izvršavaju na Windows mašini ili mašinama pod drugim OS (Sun Solaris, IBM ili Linux).



Slika 8. WCF aplikacije komuniciraju sa drugim WCF-based aplikacijama ili sa aplikacijama razvijenim na drugim Web servis platformama

Da bi obezbedio više od osnovne komunikacije, WCF implementira Web servis tehnologije definisane WS-* specifikacijom.

Ove specifikacije su u originalu definisali Microsoft, IBM i drugi proizvođači zajedno, a odnose se na nekoliko različitih oblasti, uključujući osnovnu razmenu poruka, sigurnost, stabilnost, transakcije i rad sa servisnim metapodacima.

Tabela 4. WCF implementira niz Web servis standarda

<i>Sigurnost:</i>	<i>Pouzdanost:</i>	<i>Transakcije:</i>	<i>Metadata:</i>
WS sigurnost WS pouzdanost WS bezbedonosna komunikacija	WS pouzdana razmena poruka	WS koordinacija WS atomska transakcija (all-or-nothing)	WSDL, WS-Policy, WS razmena meta podataka, WS discovery
Messaging: SOAP, WS-Addressing, MTOM			

Grupisane po funkcijama, ove specifikacije su:

- **Messaging:** SOAP je osnovni protocol za Web servise, koji definiše osnovni okvir koji sadrži zaglavlje i telo. WS-Addressing definiše dodatke za SOAP zaglavlje za adresiranje SOAP poruka
- **Message Transmission Optimization Mechanism (MTOM)** definiše optimizovan format za SOAP poruke bazirane na XML binarnoj Optimized Packaging (XOP) specifikaciji.

- **Meta podaci:** Web Services Description Language (WSDL) definiše standardni jezik za specifikaciju servisa i raznih aspekata kako web servisi mogu da budu korišćeni. WS-Policy omogućava specifikaciju još dinamičnijih aspekata ponašanja web servisa koji mogu da se opisu u WSDL. WS razmena meta podataka omogućava klijentu da zahteva opisne informacije o servisu, kao što je njegov WSDL. WCF podržava i WS-Discovery. Ovaj protokol omogućava aplikaciji da pronađe servis bilo gde u lokalnoj mreži.
- **Sigurnost:** WS sigurnost, WS pouzdanost i WS bezbedna komunikacija definišu dodatke SOAP porukama za obezbeđenje autentikacije, integriteta i zaštite podataka i drugih sigurnosnih mera.
- **Pouzadnost:** WS pouzdana razmena poruka definiše dodatke SOAP zaglavlju koji omogućava pouzdanu end-to-end komunikaciju.
- **Transakcije:** izgrađene na WS koordinaciji i WS atomskoj transakciji omogućavaju potvrdu transakcija u dve faze sa SOAP razmenom podataka.
- Takođe, WCF podržava **REST**. Umesto da šalje SOAP poruke preko HTTP, RESTful komunikacija se obavlja direktno preko HTTP ugrađenih glagola: GET, POST i dr.

Interoperabilnost sa ranijim Microsoft tehnologijama

Mnogi Microsoft klijenti su investirali ogromna sredstva u mnoge .NET Framework tehnologije koje danas obuhvata WCF. Zaštita uloženog je osnovni cilj razvojnih dijanera WCF-a. WCF ne narušava postojeću arhitekturu, tako da nema dodatnih zahteva koje organizacija treba da primeni na postojeće aplikacije [35].

5.7.1.3 Eksplicitna podrška za servisno-orijentisani razvoj.

Kreiranje servisno-orijentisanih aplikacija je postao standard.

Razmišljanje o aplikacijama koje pružaju i koriste servise je relativno nova pojava. Ono što je novo kod WCF je jasan fokus na servise, a ne na objekte.

Postoje četiri načela kojih se pridržavaju kreatori WCF-a prilikom dizajniranja ove tehnologije:

- Razmena šema, ne klase. Za razliku od starijih distribuiranih objektnih tehnologija, servisi stupaju u interakciju sa klijentima samo preko strogo definisanog XML interfejsa. Ponašanja kao što su prosleđivanje kompletne klase ili metoda preko granica servisa nisu dozvoljena.
- Servisi su autonomni. Servis i klijenti komuniciraju samo preko intefejsa između njih, a po svemu drugom su nezavisni jedan od drugog. Mogu da budu napisani u različitim programskim jezicima, da se izvršavaju u različitim okruženjima, kao CLR i JVM, da se izvršavaju na različitim operativnim sistemima.
- Granice su eksplicitno postavljene. Cilj distribuiranih objektnih tehnologija kao što je COM (DCOM) je bio da udaljeni objekat prikaže što je moguće sličnijim lokalnom objektu. Dok je ovaj pristup pojednostavnio razvoj u jednom smislu, obezbeđujući zajednički programski model, istovremeno je sakrio neizbežne razlike između lokalnih i udaljenih objekata. Servisi zaobilaze ovaj problem tako što prave eksplicitne interakcije između servisa i njihovih klijenata.

Sakrivanje razmene nije cilj.

- Koristi se politika zasnovana na kompatibilnosti. Kada god je moguće, određivanje opcija koje će se koristiti između sistema treba da se bazira na mehanizmima definisanim u WSDL i WS-Policy. Mogućnost da klijent konzumira neki servis, bazirana je na preseku elementa koje podržava servis i onih koje podržava klijent [35].

5.7.2 Osnovni koncepti WCF-a

5.7.2.1 EndPoint (Krajnja tačka)

WCF servis je program koji izlaže kolekciju EndPoint-a. Svaki od EndPoint-a je portal za komunikaciju sa spoljnim svetom.

Sva komunikacija WCF-a se obavlja preko EndPoint-a.

EndPoint se sastoji iz tri komponente:

Adresa – najčešće URL koji specificira adresu gde hostuje WCF. Klijent koristi URL da se poveže sa servisom.

Binding – (vezivanje). Binding opisuje kako klijent komunicira sa servisom. Postoje različiti protokoli kojima je moguća komunikacija između WCF i klijenta.

Binding ima nekoliko karakteristika:

- Transport koji definiše osnovni protokol koji će se koristiti, kao npr. HTTP, Named Pipes, TCP, MSMQ
- Encoding: text, binary ili Message Transmission Optimization Mechanism (MTOM). MTOM omogućava prenos velikih formata (preko 64K).
- Protokol koji definiše informacije koje će se koristiti za povezivanje kao što je sigurnost, transakcije ili bezbedonosne poruke.

U tabeli je prikazana lista nekih od protokola koji su omogućeni kod WCF povezivanja:

Tabela 5. Protokoli za WCF povezivanje

Binding (povezivanje)	Opis
BasicHttpBinding	Osnovna web servis komunikacija. Bez sigurnosnih mehanizama u osnovnoj postavci.
WSHttpBinding	Web servis sa *.WS podrškom. Podržava transakcije.
WebHttpBinding	REST-stil povezivanje. Odgovaraju HTTP zahtevima umesto SOAP porukama.
WSDualHttpBinding	Web servis sa duplex ugovorima i podrškom za transakcije.
WSFederationHttpBinding	Web servis sa federacijskom sigurnošću. Podržava transakcije.
MsmqIntegrationBinding	Komunikacija direktno sa MSMQ aplikacijama. Podržava transakcije.

NetMsmqBinding	Komunikacija između WCF aplikacija preko redova. Podržava transakcije.
NetNamedPipeBinding	Komunikacija između WCF aplikacija na istom računaru. Podržava duplex ugovore i transakcije.
NetPeerTcpBinding	Komunikacija između računara preko P2P servisa. Podržava duplex ugovore.
NetTcpBinding	Komunikacija između WCF aplikacija između računara. Podržava duplex ugovore i transakcije.

Ugovor (Contract) - Ugovor je kolekcija operacija koja specificira čime će krajnja tačka da komunicira sa spoljnim svetom. Često se u ugovoru navodi naziv imenskog prostora tako da klijentska aplikacija bude svesna svesna operacija koje su joj izložene. Svaka operacija je jednostavna razmena podataka po određenom paternu: jednosmerna, request/reply ili duplex.

Primer postavke adrese, vezivanja i ugovora:

```
<endpoint address="localhost/wcfBloodon/RestHealthGuide.svc" binding="webHttpBinding"
contract="wcfBloodon.IRestHealthGuide" behaviorConfiguration="web"/>
```

5.7.2.2 Povezivanje i ponašanje (Binding i Behavior)

Povezivanje (Binding) govori o tome kako će klijent komunicirati sa servisom. Konfiguriše se u konfiguracionom fajlu (.config fajl).

5.7.2.3 Ugovori i servisni hostovi (Contract i Service host)

U WCF-u svi servisi su izloženi kao ugovori (Contracts). Ugovor je platformski neutralan i standardni način za opisivanje šta servis radi. Postoje četiri tipa ugovora u WCF: ServiceContract, DataContract, MessageContract, FaultContract.

Servisni host je objekat sa svrhom hostovanja servisa i registracije krajnjih tačaka. Sadrži servisnu konfiguraciju krajnjih tačaka, prihvata razna podešavanja i startuje osluškivače za nadolazeće zahteve. *System.ServiceModel.ServiceHost* je imenski prostor ovog objekta.

5.7.3 Razlike između WCF i Web servisa

Web servis je deo WCF-a. WCF nudi više fleksibilnosti i portabilnosti za razvoj servisa za razliku od web servisa. Na sledećoj tabeli prikazane su najznačajnije razlike.

Tabela 6. Razlike između web servisa i WCF

	Web servis	WCF
Hostovanje	Može da hostuje na IIS	Može da hostuje na IIS, Windows activation servisu, Self-hosting i kao Windows servis
Programiranje	[WebService] atribut se dodaje klasi	[ServiceContract] atribut se dodaje klasi
Model	[WebMethod] atribut predstavlja metod koji je izložen ka klijentu	[OperationContract] atribut predstavlja metod koji je izložen ka klijentu
Operacije	Jednosmerno, Request-Response su različite operacije podržane u web servisu	Jednosmerno, Request-Response, Duplex su različite operacije podržane u WCF
XML	System.Xml.serialization je imenski prostor za serijalizaciju	System.Runtime.Serialization je imenski prostor za serijalizaciju
Encoding	XML 1.0, MTOM(Message Transmission Optimization Mechanism), DIME, Custom	XML 1.0, MTOM, Binary, Custom
Transport	HTTP, TCP	HTTP, TCP, Named pipes, MSMQ, P2P
Protokoli	Sigurnost	Sigurnost, Pouzdanost, Transakcije

5.8 Standardi i okviri za razvoj web servisa

Web servisi bazirani na XML-u ili JSON-u najčešće se implementiraju pomoću Java 2 Platforme (J2EE) (Sun Microsystems) i Microsoft.NET (Microsoft Corporation).

5.8.1 J2EE i Microsoft.NET pristup web servisima

Web servisi moraju da budu pouzdani, dostupni, skalabilni i moraju da se izvršavaju na zadovoljavajući način. Ovi zahtevi se ne razlikuju od zahteva bilo koje druge aplikacije.

J2EE i .NET predstavljaju evoluciju postojeće serverske tehnologije koja se koristi za razvoj aplikacija. Ranije verzije obe ove tehnologije nisu omogućavale razvoj web servisa. Sa pojavom i razvojem web servisa, obe strane su repozicionirale svoja rešenja i prilagodila ih izgradnji web servisa.

Zajednička vizija i J2EE i .NET-a jeste da treba da postoji XML interoperabilnost, odnosno da treba pisati aplikativna rešenja koja će se izvršavati u okruženjima koja će obezbeđivati različite servise.

5.8.1.1 J2EE

Java 2 Platform, Enterprise Edition (J2EE) je dizajnirana da pojednostavi složene probleme kod razvoja i održavanja višeslojnih poslovnih rešenja. J2EE je standard i

rezultat je velike tehnološke inicijative predvođene kompanijom Sun Microsystems.

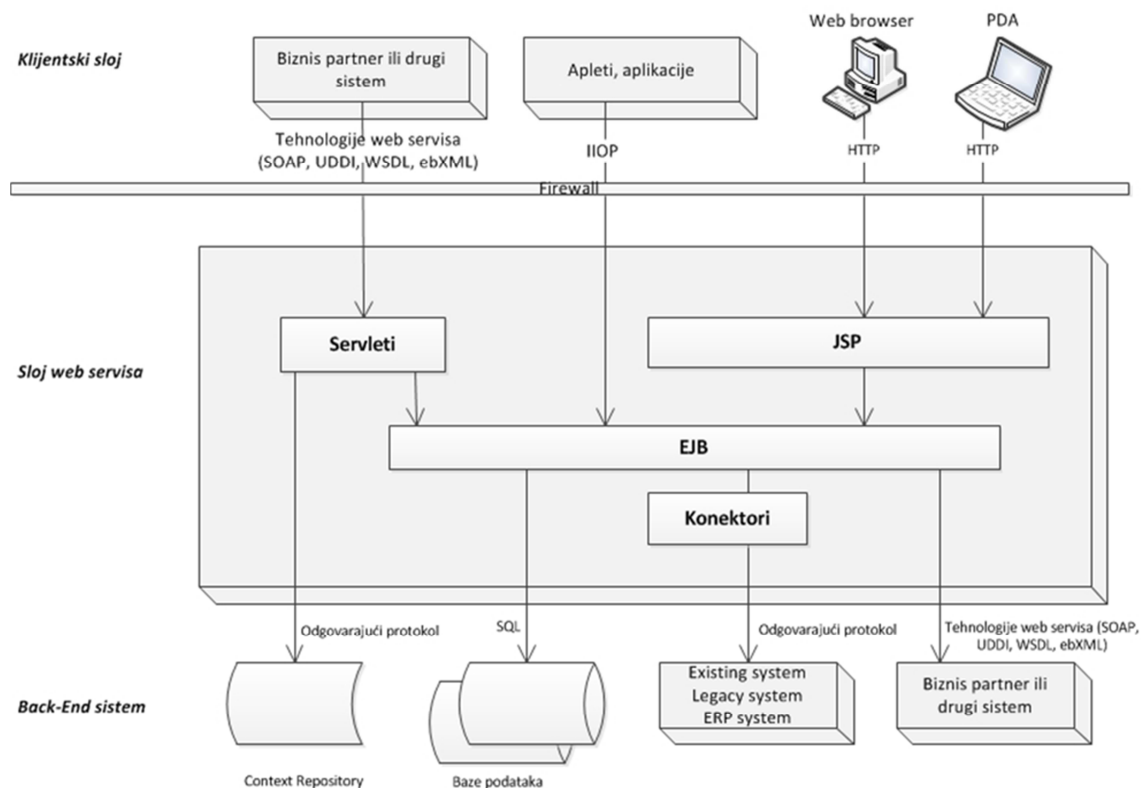
J2EE je standard, a ne proizvod. Ovaj standard opisuje “ugovore” između aplikacija i okruženja u kojima se izvršavaju. Dok sve strane poštuju iste standarde aplikacija može da bude razvijena u različitim okruženjima. Cilj J2EE jeste da pruži korisnicima izbor alata i proizvoda kao i da podstakne razvoj zdrave konkurencije. Ovo će biti moguće postići ako se svi proizvođači prilagode J2EE standardima. Da bi obezbedila kompatibilnost, kompanija Sun je radila na razvoju standarda zajedno sa drugim proizvođačima eBusiness platformi: BEA, IBM i Oracle.

J2EE arhitektura je bazirana na Java programskom jeziku.

5.8.1.2 J2EE i Web servisi

J2EE je izvorno bila arhitektura za razvoj serverskih aplikacija u Java programskom jeziku. U istoj arhitekturi mogu da se razvijaju tradicionalni web servisi, softverske komponente ili pakete aplikacije. J2EE je nedavno proširena za razvoj web servisa baziranih na XML-u. Ovi web servisi mogu da komuniciraju sa drugim web servisima koji mogu i ne moraju da budu razvijeni pod J2EE standardima.

J2EE razvoj web servisa je predstavljen sledećom slikom:



Slika 9. Razvoj web servisa sa J2EE [36]

J2EE aplikacija hostuje u web servis okruženju (Container) koji obezbeđuje servise za poslovne aplikacije kao što su transakcije, zaštita i postojanje servisa.

Poslovni sloj obezbeđuje poslovanje i logički deo operacija. J2EE aplikacije, poslovni sloj je razvijen uz pomoć Enterprise JavaBeans (EJB) komponenti.

Ovaj sloj komunicira sa bazom podataka preko Java Database Connectivity (JDBC) ili SQL/J, ili postojećeg sistema uz pomoć Java Connector Architecture (JCA). Takođe, može da se poveže sa partnerskim aplikacijama pomoću web servis tehnologija (SOAP, UDDI, WSDL, ebXML) preko Java API za XML (JAX APIs).

Poslovni partneri mogu da se povežu sa J2EE aplikacijama preko web servis tehnologija. Servlet, koji predstavlja "request/response" Java orijentisani objekat, koji prihvata zahteve za web servisima od drugih sistema. Servlet koristi JAX API da izvrši operacije web servisa.

Tradicionalne aplikacije ili apleti direktno se povezuju sa EJB slojem preko Internet Inter-ORB protokola (IIOP), a ne preko web servisa jer nema potrebe za kolaboracijom obzirom da su pisani istom tehnologijom i pod istim standardima.

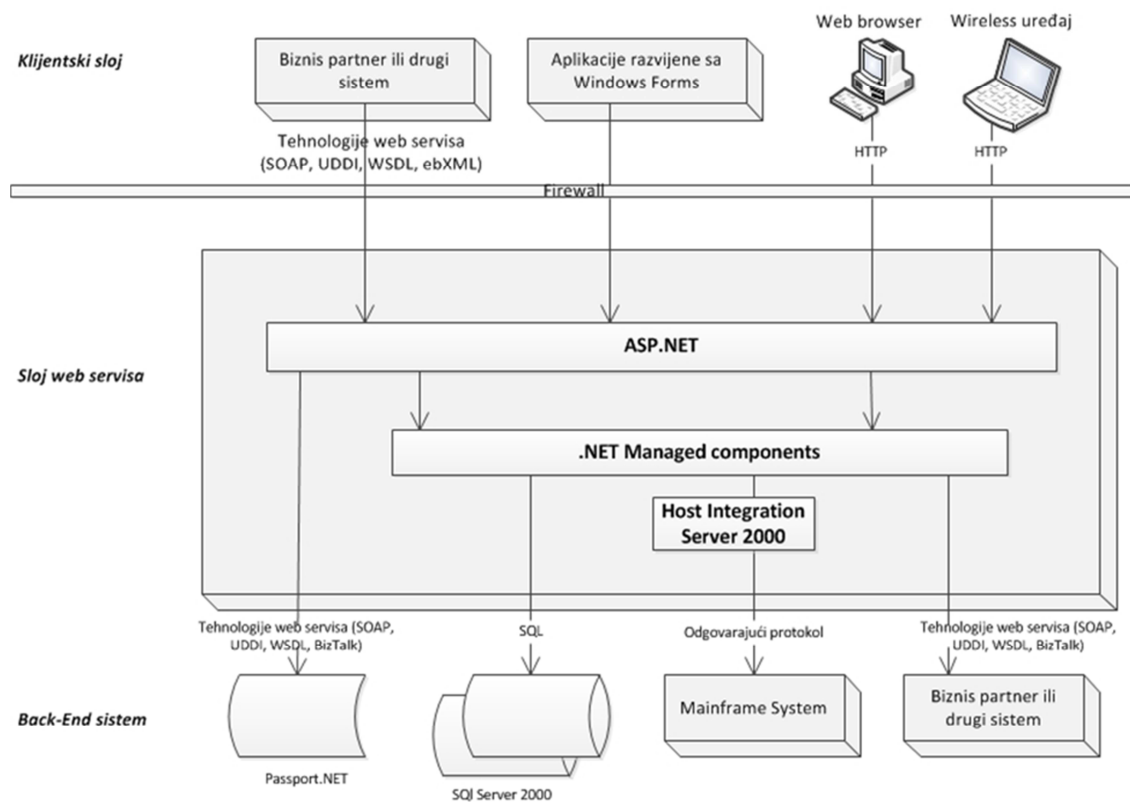
Web browser-i i bežični uređaji se konektuju na JavaServer Pages (JSPs) koje renderuju korisnički interfejs u HTML, XHTML ili WML, a preko EJB komponenti dolaze do baze podataka ili web servisa.

5.8.1.3 Microsoft-ova .NET platforma

Microsoft.NET je novi okvir koji omogućava organizacijama da razvijaju pametne poslovne aplikacije, uključujući Windows aplikacije, aplikacije bazirane na web-u. .NET se može koristiti za razvoj sistema koji se sastoje od međusobno povezanih servisa koji komuniciraju jedan sa drugim preko Interneta.

Pored toga, .NET se može koristiti za kreiranje aplikacija koje se izvršavaju na uređajima kao što su ručni računari i mobilni telefoni [6].

.NET je proizvodna strategija dok J2EE predstavlja standard po kojem su proizvodi napisani. Microsoft.NET je prerada Windows DNA platforme koju je ranije razvio Microsoft za razvoj poslovnih aplikacija. Windows DNA sadrži mnoge poznate tehnologije koje i danas postoje u produkciji kao što su Microsoft Transaction Server (MTS) i COM+, Microsoft Message Queue (MSMQ) i Microsoft SQL Server baza podataka. Novi .NET Framework zamenjuje ove tehnologije, a uključuje sloj web servisa kao i naprednu jezičku podršku. Na sledećoj slici je prikazan programerski model za razvoj web servisa sa Microsoft.NET.



Slika 10. Razvoj web servisa sa Microsoft.NET [36]

.NET aplikacije se nalaze u okviru web servis okruženja koji obezbeđuje servise neophodne za poslovne aplikacije, kao što su transakcije, zaštita ili servisi za poruke.

Poslovni sloj .NET aplikacija je razvije uz pomoć .NET komponenti. Ovaj sloj obezbeđuje procesiranje i logički deo operacija. Povezuje se sa bazom podataka uz pomoć Active Data Objects (ADO.NET) i pomoću servisa koje obezbeđuje Microsoft Host Integration Server 2000 (COM Transaction Integrator). Takođe, sa drugim poslovnim aplikacijama i sistemima može da se poveže preko web servis tehnologija (SOAP, UDDI, WSDL).

Poslovne aplikacije (partneri) mogu da se povežu sa .NET aplikacijama preko web servis tehnologija (SOAP, UDDI, WSDL, BizTalk).

Tradicionalne aplikacije, klijenti, web browser-i, bežični uređaji povezuju se sa Active Server Pages (ASP.NET) koje renderuju korisnički interfejs u HTML, XHTML ili WML.

Tabela 7. Prikaz analogija kod J2EE i .NET

Svojstvo	J2EE	.NET
Tip tehnologije	Standard	Proizvod
Middleware Vendors	30+	Microsoft
Interpreter	JRE	CLR

Dinamičke web strane	JSP	ASP.NET
Middle-Tier Components	EJB	.NET Managed Components
Pristup bazi podataka	JDBC SQL/J	ADO.NET
SOAP, WSDL, UDDI	DA	DA
Implicit middleware (load-balancing, etc)	DA	DA

5.8.1.4 .NET Framework

Microsoft.NET nudi jezičku nezavisnost i jezičku interoperabilnost. Ovo je jedna od najznačajnijih karakteristika .NET platforme. Jedna .NET komponenta može da bude napisana jednim delom u VB.NET (.NET verzija Visual Basic jezika), a drugim delom u C#, Microsoft-ov novi objektno orijentisani jezik.

.NET framework se sastoji iz tri komponente:

Development languages and tools. Jezici za razvoj koji omogućavaju pisanje .NET programa uključujući C#, Visual Basic.NET (VB.NET) i Managed C++. Microsoft ima i alat za brzi razvoj aplikacija (RAD – Rapid Application Development) koji se naziva Visual Studio.NET, koji omogućava razvoj programa u integrisanom razvojnom okruženju (IDE-integrated development environment).

Common Language Runtime (CLR) upravlja kodom koji se izvršava i obezbeđuje servise kao što je upravljanje memorijom, upravljanje nitima (thread – koje omogućavaju izvršavanje više zadataka istovremeno) i daljinsko upravljanje (remoting – koji omogućava objektima iz jedne aplikacije da komuniciraju sa objektima u drugoj aplikaciji). CLR ima i sigurnost i strogu tačnost izvršnog koda kako bi se obezbedilo da ne dođe ni do kakvog ometanja.

CLR je Microsoft-ov posrednik između .NET programerskog izvornog koda i izvršnog hardverskog.

CLR predstavlja analogiju sa JRE kod Java jezika, koji interpretira IL kod.

Framework Base Class Library je ogromna kolekcija koda koji je napisan u Microsoft-u i koji se može koristiti u programima. Framework Base Class Library sadrži kod koji, pored mnogih funkcija, omogućava razvoj Windows aplikacija, pristup direktorijumima i fajlovima na disku, konekciju na bazu i vraćanje informacija, kao i slanje i primanje podataka na mreži.

Izvorni kod se prevodi u Microsoft Intermediate Language (MSIL ili IL).

IL je jezički neutralan i predstavlja analogiju sa Javinim bytecode-om.

Pošto računar može da pokrene program napisan na njegovom jeziku, a to je mašinski kod (niz binarnih brojeva, nula i jedinica), stoga se IL interpretira u mašinski kod, odnosno izvršni jezik. Ovaj korak se izvršava automatski pomoću kompajlera

Just In Time (JIT).

JIT kompilacija se obavlja samo pri prvom pokretanju programa, a rezultujući mašinski kod se automatski memoriše. Pri sledećem pokretanju ponovo se koristi prethodno memorisani mašinski kod [37].

5.8.2 Komparacija J2EE i .NET platforme

J2EE promoviše programski jezik Java kao što i komponente u okviru J2EE (kao EJB komponente i servleti) moraju da budu napisane u Javi. Da bi se koristio J2EE bar jedna od poslovnih komponenti mora da bude napisana u Javi, a drugi jezici mogu da se nađu u okviru J2EE rešenja preko web servisa, CORBA, JNI ili JCA, ali ti drugi jezici ne mogu da budu pomešani sa Java programski jezikom. Teorijski JVM bytecode je jezički neutralan, ali u praksi bytecode se koristi samo sa Javom.

Poređenja radi, .NET podržava razvoj na bilo kojem jeziku koje podržava Microsoft-ov CRL. Sa izuzetkom Jave, svi ključni jezici su podržani. Prevedeni u IL postaju i ponašaju se kao jedan zajednički jezik. Jedna .NET komponenta može da bude napisana na više jezika.

Višejezička podrška koju je uveo Microsoft sa CRL je značajna poslovna inovacija i tu se jasno oslikava prednost .NET-a u odnosu na J2EE. Međutim, postoje situacije u kojima pojedinac treba da odluči šta je najbolje za jedno poslovno rešenje. Ukoliko se radi o dizajnu sa više jezika, CRL može da predstavlja lošije dizajnersko rešenje iz više razloga:

Mnogi postojeći sistemi su interno međusobno povezani. Presecanje postojećih veza predstavlja veliki veliki rizik i zahteva utrošak velike količine vremena. U ovom slučaju važi poznata premisa: “ako nije pokvareno, ne popravljaj”.

Pretpostavlja se da kombinacija više jezika u okviru CRL-a može da dovede do nesređenosti koda i teškog održavanja. Ako je reč o aplikaciji koja je pisana na više jezika, da bi se održavala ili nadograđivala potrebno je pronaći eksperte za više jezika, što svakako za sobom vuče i veće troškove održavanja.

Sa kombinacijom više jezika, programeri nisu u prilici da razmenjuju iskustva. Možda dolazi do povećanja produktivnosti individue, ali komunikacija u timu kao i timska produktivnost opada.

Argumenti koji podržavaju obe platforme:

- Bez obzira o kojoj platformi se radi, novi programeri treba da prođu obuku (Java za J2EE, OO za .NET);
- Web servisi mogu da se razvijaju na obe platforme;
- Obe platforme nude jeftina sistemska rešenja, kao jBoss/Linux/Cobalt za J2EE ili Windows/Win32 hardver za .NET;
- Skalabilnost oba rešenja su teorijski neograničena.

Argumenti za .NET, a protiv J2EE

- .NET promoviše Microsoft;
- .NET je razvio rešenja za web servise nešto pre J2EE i u tom smislu ima prednosti;
- .NET pruža ogromne mogućnosti sa Visual Studio.NET;
- .NET ima jednostavan programerski model koji omogućava veliku produktivnost;
- .NET pruža jezičku neutralnost kada se razvijaju nova poslovna rešenja, dok se kod J2EE drugi jezici tretiraju kao zasebne aplikacije;
- .NET izvlači korist od tesne povezanosti i prepletenosti sa operativnim sistemom na kojem se nalazi.

Argumenti za J2EE, a protiv .NET

- J2EE promoviše cela industrija;
- J2EE je dokazana platforma sa nekoliko novih API za web servise. .NET je unapređena i predstavlja rizik kao i svaka prva tehnološka generacija;
- Postojeći J2EE kod može lako da se prevede u J2EE web servis sistem, dok to nije slučaj sa prevođenjem Microsoft DNA u .NET;
- .NET web servisi nisu interoperabilni sa trenutnim industrijskim standardima. Njihov BizTalk framework ima odgovarajuće SOAP ekstenzije ali ne podržava ebXML;
- J2EE je mnogo napredniji programerski model, prikladan za iskusne programere koji razvijaju napredne objektne modele i performanse;
- J2EE omogućava iskorišćenje prednosti postojećeg hardvera;
- J2EE omogućava neutralnost platforme, uključujući i Windows;
- J2EE dozvoljava korišćenje bilo koje OS: Windows, UNIX ili mainframe;
- J2EE omogućava pisanje koda u Javi, što je bolje nego u C# zbog razmene iskustava;
- Prema Gartneru, postoji 2,5 miliona Java programera. Predviđa se još veći rast [38].

5.9 Programska okruženja za razvoj web servisa

5.9.1 .NET proizvodi i servisi

Nova verzija Visual Studio-a je u upotrebi kao Visual Studio.NET (VS.NET), integrisano razvojno okruženje za .NET platformu.

VS.NET je Microsoft-ov alat za brzi razvoj aplikacija (RAD-Rapid Application Development). VS.NET je integrisano razvojno okruženje koje se može koristiti za kreiranje velikog broja tipova .NET programa. On omogućava i unošenje programa, njihovo kompajliranje i izvršavanje uz pomoć Windows grafičkog okruženja jednostavnog za korišćenje. VS.NET omogućava prolazak kroz svaku liniju koda u toku izvršavanja programa – debaging.

VS.NET se može koristiti za razvoj više tipova aplikacija:

Windows aplikacija, ASP.NET aplikacija, WCF aplikacija i ASP.NET web servisa [37].

Ima podršku za sve programske jezike kao i Visual Studio, osim Jave.

Visual Studio.NET obezbeđuje podršku za novi Microsoft-ov programski jezik C# koji je semantički sličan Javi sa nekoliko sintaksnih razlika.

5.9.2 Eclipse

Eclipse Project je izorno kreirao IBM u Novembru, 2001. godine. Eclipse fondacija je ustanovljena u januaru, 2004. godine kao nezavisna, neprofitabilna organizacija.

Danas Eclipse zajednicu sačinjavaju pojedinci i organizacije iz raznih oblasti softverske industrije koji rade [41].

Eclipse fondacija upravlja razvojem Eclipse-a. Fondacija obezbeđuje servise zajednici, ali ne zapošljava programere koji rade na razvoju Eclipse-a. Programeri su najčešće zaposleni u drugim organizacijama ili rade volonterski, u slobodno vreme na razvoju Eclipse-a.

Eclipse je Java-orijentisana, "open source" razvojna platforma. Sama po sebi čini jednostavan framework-okvir sa setom servisa za razvoj aplikacija.

"Open source" softver je takav softver koji se izdaje sa licencom, a sa namerom da se obezbede određena prava korisnicima. Jedno od osnovnih prava je, naravno, da izvorni kod treba da bude dostupan korisnicima koji ga mogu modifikovati.

Eclipse dolazi u paketu sa setom plug-in-ova, uključujući i dobro poznat Java development Tools (JDT).

Eclipse Web Tools Platform (WTP) jeste dodatak Eclipse platformi sa alatima koji omogućavaju razvoj Web i Java EE aplikacija. Sadrži izvorni kod i grafičke editore za razne jezike, pomoćnike i ugrađene aplikacije koje olakšavaju razvoj softvera, kao i alate i API koji pomažu razvoj, testiranje i pokretanje aplikacija.

Iako je većina programera prilično zadovoljna koristeći Eclipse kao Java orijentisano okruženje (IDE), Eclipse nudi i Plug-in Development Environment (PDE) koji je prvenstveno od interesa onima koji žele da prošire mogućnosti Eclipse-a. U Eclipse-u je gotovo sve u plug-in-ovima.

Iako je Eclipse pisan u Javi i podržava prvenstveno Javu, moguće je proširiti platformu plug-in-ovima za C/C++ i COBOL. Eclipse framework može da bude i polazna osnova za druge tipove aplikacija, koje nemaju veze sa softverskim razvojem, kao što je "content management system" (CMS).

Od nastanka Eclipse-a do danas, svake godine u junu biva plasirana nova verzija Eclipse-a. Najnovija verzija nosi naziv Juno, od 27.06.2012. godine. Prethodna verzija (22.06.2012.) – 3.7 nosi naziv Indigo.

5.10 Android i programska okruženja za razvoj Android aplikacija

5.10.1 Android

Android je softverski stek koji sadrži operativni sistem, middleware i ključne aplikacije za mobilne uređaje kao što su pametni telefoni ili tablet računari. Razvio ga je The Open Handset Alliance na čelu sa Google-om, a baziran je na Linux-ovom kernelu. Android OS je otvoren – open source (pod Apache licencom), a razvijaju ga programeri širom sveta.

Ovaj operativni sistem podržava mnoge hardverske komponente, npr. kameru, WiFi komunikacioni čip, mobilni komunikacioni čip, Bluetooth, ekran osetljiv na dodir, NFC (Near Field Communications) i dr.

Android SDK (Software Development Kit) obezbeđuje alate i API (Application Program Interface) neophodne za razvoj aplikacija na Android platformi u programskom jeziku Java. Android API sadrži mnoge funkcije i klase koje su u stanju da kontrolišu funkcionalnosti mobilnih uređaja.

Android projekat se sastoji iz fajlova sa izvornim kodom Android aplikacije. Android SDK alati omogućavaju jednostavnu izradu novog Android projekta sa setom osnovnih projektnih direktorijuma i fajlova.

Iako je većina Android aplikacija pisana u Javi, postoje mnoge razlike između Java API i Android API, a uz to Android ne koristi Java virtuelnu mašinu (JVM) nego Dalvik. Java bytecode se ne izvršava – Java klase su kompajlirane u Dalvik izvršne fajlove i izvršavaju se na Dalvik virtuelnoj mašini dizajniranoj specijalno za Android.

5.10.2 Programski jezik Java

Android aplikacije se razvijaju na programskom jeziku Java i za sada je to jedina opcija za razvoj nativnih Android aplikacija.

Java je jako popularan programski jezik kojeg je razvio Sun Microsystems (sada vlasništvo kompanije Oracle). Razvijena je posle jezika C i C++ i inkorporira mnoge moćne funkcije koje poseduju ova dva jezika, a uz to rešava neke njihove nedostake.

Kaže se da su programski jezici moćni koliko su moćne njihove biblioteke. One su te koje programerima omogućavaju razvoj aplikacija.

Glavna odlika Jave jeste to što jednom napisan kod može da se izvršava na bilo kojoj platformi. Proces se odvija tako što:

- Programeri napišu izvorni kod u Javi
- Java kod se kompajlira u bytecode, međuplatformski jezik, između izvornog i mašinskog jezika
- Kada je kod spreman za izvršenje, JRE (Java Runtime Environment) interpretira bytecode i izvršava ga u realnom vremenu.

Neke od Javinih važnih osobina su:

- Jezik je jednostavan za učenje
- Dizajniran je da bude nezavisan od platforme i siguran koristeći VM
- Objektno je orijentisan

Android se u potpunosti oslanja na ove ključne osobine. Android SDK uključuje mnoge standardne Java biblioteke (strukturne biblioteke, matematičke biblioteke, grafičke i mrežne biblioteke i dr.), kao i specijalne Android biblioteke [39].

5.10.3 Sigurnost izvršenja Java koda

Obzirom da se Java aplikacije izvršavaju u okviru Dalvik VM, kojeg možemo posmatrati kao balon ("bubble"), one su izolovane od ostatka uređaja, odnosno hardvera.

VM može da upravlja izvršenjem koda na siguran način za razliku od drugih jezika koji "operišu" direktno sa mašinskim jezikom.

Android platforma je otišla čak i korak dalje. Svaka Android aplikacija se izvršava na Linux OS koristeći zaseban nalog i svoju sopstvenu instancu dalvik VM.

OS vrši nadgledanje izvršenja Android aplikacija i ako izvršenje nije regularno (ako se troši previše memorije, zauzeće bitnih resursa i sl), aplikacije će biti ugašene.

Aplikacije mogu međusobno da komuniciraju preko specijalno definisanih protokola [40].

5.10.4 Kompajliranje Java koda

Kao i mnogi jezici, tako i Java mora da se kompajlira, čak i ako se ne kompajlira do mašinskog koda. To znači da programeri koji razvijaju Android projekte moraju da izvrše pakovanje projekta kako bi se mogao preneti na uređaj.

Eclipse razvojno okruženje (sa Android Development Plug-in-om) omogućava automatsko kompajliranje. To znači da svaki put kada se sačuva promena izvršena na fajlu, Eclipse rekompajlira promene u projektu. Automatski su vidljive i načinjene greške.

Eclipse, takođe, interpretira Javu dok se unosi kod, omogućavajući formatiranje i greške u momentu unosa koda.

"Build" proces, kod regularnih java projekata, rezultira fajlom sa .jar ekstenzijom – JavaARchive.

Android aplikacije uzimaju .jar fajl i pakuju ga za prenos na Android uređaje kao AndroidPacKage fajl sa ekstenzijom .apk.

Ovakav paket ne sadrži samo kompajlirani Java kod već i mnoge resurse, kao što su stringovi, zvukovi, slike i druge fajlove koji su potrebni aplikaciji. Tu je i AndroidManifest fajl koji služi za definisanje konfiguracionih detalja o aplikaciji [40].

5.10.5 Programska okruženja za razvoj Android aplikacija

Za razvoj Android aplikacija mogu da se koriste Netbeans ili Eclipse. Oba Integrated Development Environment (IDE) su besplatna i jednostavna za korišćenje. Eclipse je nastao pre Netbeans-a.

5.10.5.1 Eclipse

Eclipse je veoma popularno razvojno okruženje (IDE) za Javu i omiljeno okruženje za razvoj Android aplikacija. Besplatno je i dostupno za Windows, Mac i Linux OS.

Sa Android SDK alatom za Eclipse (r20 od juna, 2012. godine) moguće je jednostavno kreirati sopstvene biblioteke i referencirati ih u okviru Android aplikacija.

Android biblioteke mogu da sadrže Java klase, resurse i druge informacije o projektu. Eclipse je preporučeno rešenje za razvoj Android aplikacija od strane Google-a.

5.10.5.2 NetBeans

NetBeans se odnosi na platformu za Java desktop aplikacije kao i na Integrated Development environment (IDE) za razvoj aplikacija u raznim jezicima kao što su: Java, JavaScript, PHP, Python, C, C++ i drugi.

NetBeans IDE je napisan u Javi, a može da se izvršava na Windows, Mac OS, Linux, Solaris i drugim platformama koje podržavaju JVM. NetBeans IDE omogućava razvoj svih tipova Java aplikacija – Java SE, Java ME, EJB i mobilne aplikacije. NetBeans platforma omogućava aplikacijama da se razvijaju iz softverskih komponenti zvanih moduli. Aplikacije bazirane na NetBeans platformi mogu da se dalje razvijaju od strane trećih lica.

NetBeans se pojavljuje 1996. godine pod nazivom Xelfi (igra reči od Delphi), kao studentski projekat pri Fakultetu za matematiku i fiziku, Charles Univerziteta u Pragu, a 1999. godine ga kupuje Sun Microsystems.

Sun je sledeće godine učinio NetBeans "open source" projektom. Od 2010. godine Sun, a time i NetBeans preuzima Oracle.

Trenutna verzija je NetBeans IDE 7.0.1, iz avgusta 2011. godine.

Među mnogobrojnim funkcijama ističu se:

1. Upravljanje korisničkim interfejsom (meniji, alatke);
2. Korisničko podešavanje;
3. Upravljanje čuvanjem podataka;
4. Upravljanje okvirima;
5. Čarobnjaci;
6. NetBeans Visual biblioteka;
7. Integrisani razvojni alati.

Za razvoj Android aplikacija potrebno je preuzeti Android plugin za NetBeans – nbandroid.

6 Realizovano rešenje

U tekstu koji sledi dat je pregled rešenja kojeg smo realizovali i to serverske podrške za Android aplikaciju i klijentske Android aplikacije za Health Care sistem.

Klijentska aplikacija omogućava korisnicima da popune upitnik za dobrovoljno davanje krvi, da izvrše registraciju, a potom proslede svoje odgovore do servera gde će se izvršiti njihova obrada (originalni upitnik se nalazi u poglavlju Prilozi na kraju rada). Konačna informacija o tome da li korisnik može da poseti Zavod radi dobrovoljnog davanja krvi, šalje se nazad do klijenta putem servisa.

Aplikacija za dobrovoljne davaoce krvi je samo jedna od ideja koju smo realizovali i koja prikazuje kako je moguće približiti korisniku brojne usluge, dobiti odgovarajuće informacije od korisnika u cilju epidemiološkog praćenja zdravlja, smanjenja redova čekanja koji nastaju usled vremena potrebnog za ispunjavanje raznih formulara u zdravstvenim ustanovama (sistematski pregledi, anekete o zdravstvenom stanju, pregledi za vozače i sl.).

6.1 Servisna podrška Android aplikaciji

Za realizaciju serverske aplikacije koristili smo Visual Studio.NET 2010 razvojno okruženje, WCF programsku platformu i izvršno okruženje za razvoj i konfigurisanje mrežno-distribuiranih servisa i programski jezik C#.

Za skladištenje podataka smo koristili Microsoft SQL Server 2008.

Za programsko okruženje VS.NET smo se opredelili, osim brojnih prednosti, zbog široke rasprostranjenosti .NET tehnologije među zdravstvenim institucijama u našoj zemlji. Obzirom da je istraživanje bazirano na podršci i unapređenju zdravstvenog sistema i pružanja zdravstvenih usluga u Srbiji, smatrali smo da je korisno iskoristiti postojeće infrastrukturne potencijale u zdravstvenom sistemu.

6.1.1 Implementacija WCF servisa

Servisni ugovori i operativni ugovori (Service i Operation Contracts) su definisani u Interfejsu IRestHealthGuide. Interfejs je označen atributom [ServiceContract], a metodi atributom [OperationContract].

Klasa *IRestHealthGuide.cs*

```
using System.Collections.Generic;
using System.ServiceModel;
using System.ServiceModel.Web;

namespace wcfBloodon
{
    [ServiceContract]
    public interface IRestHealthGuide
    {
        [OperationContract]
        [WebInvoke(Method = "GET",
            ResponseFormat = WebMessageFormat.Json,
            BodyStyle = WebMessageBodyStyle.Wrapped,
            UriTemplate = "getQuestions")]
        List<Question> GetQuestions();

        [OperationContract]
        [WebInvoke(Method = "POST",
            RequestFormat = WebMessageFormat.Json,
            ResponseFormat = WebMessageFormat.Json,
            BodyStyle = WebMessageBodyStyle.Bare,
            UriTemplate = "register")]
        string Register(Registration data);

        [OperationContract]
        [WebInvoke(Method = "GET",
            ResponseFormat = WebMessageFormat.Json,
            BodyStyle = WebMessageBodyStyle.Wrapped,
            UriTemplate = "getToken")]
        string getToken();

        [OperationContract]
        [WebInvoke(Method = "POST",
            RequestFormat = WebMessageFormat.Json,
            ResponseFormat = WebMessageFormat.Json,
            BodyStyle = WebMessageBodyStyle.Wrapped,
            UriTemplate = "validate")]
        string Validate(List<Question> questions, string token);
    }
}
```

Implementacija interfejsa je izvršena u klasi *RestHealthGuide* i pomoćnoj klasi *DataDAL*.

Klasa *RestHealthGuide.cs*

```
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.ServiceModel;
using System;
using System.Linq;

namespace wcfBloodon
{
    public class RestHealthGuide : IRestHealthGuide
    {
        public List<Question> GetQuestions()
        {
            return DataDAL.GetQuestions();
        }
        public string Register(Registration data)
        {
            return DataDAL.Register (data);
        }
        public string getToken()
        {
            string token = Guid.NewGuid().ToString();
            return DataDAL.saveToken(token);
        }
        public string Validate(List<Question> questions, string token)
        {
            List<Question> qa = DataDAL.GetQuestions();
            foreach (Question q in questions)
            {
                if (q.answer != q.yes)
                    return "false";
            }
            return DataDAL.saveAnswersToDatabase(questions, token);
        }
    }
}
```

Klasa *DataDAL.cs*, metod *GetQuestions ()* koji vraća pitanja iz baze podataka.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Configuration;
using System.Data.SqlClient;

namespace wcfBloodon
{
    public class DataDAL
    {
        public static List<Question> GetQuestions()
        {
            var list = new List<Question>();
            using (SqlConnection conn = new
            SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ToString()))
            using (var command = new SqlCommand("SELECT id, question, answer FROM dbo.Questions",
            conn))
            {
                conn.Open();
                using (SqlDataReader rdr = command.ExecuteReader())
                {
                    while (rdr.Read()) {
                        var q = new Question()
                        {
                            id = rdr.GetInt32(0),
                            question = rdr.GetString(1),
                            answer = rdr.GetBoolean(2)
                        };
                        list.Add(q);}
                    rdr.Close();
                } conn.Close();
            }
            return list;
        }
    }
}
```

Konfiguracija WCF projekta izvršena je u *Web.config* fajlu gde smo kao povezivanje (binding) naveli *webHttpBinding* koji odgovara REST-style povezivanju.

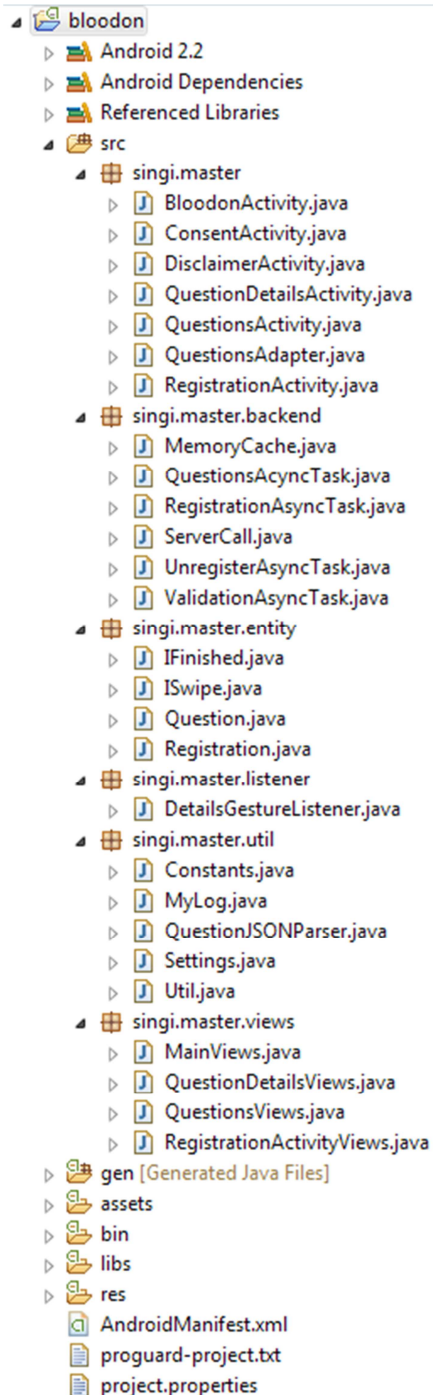
Web.config fajl

```
<configuration>
  <connectionStrings>
    <clear/>
    <add name="ConnectionString" connectionString="Data Source=*****;Initial
Catalog=health;Persist Security Info=True;User ID=*****;Password=*****"
providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <system.serviceModel>
    <services>
      <service name="wcfBloodon.RestHealthGuide">
        <endpoint address="" binding="webHttpBinding"
contract="wcfBloodon.IRestHealthGuide" behaviorConfiguration="web"/>
      </service>
    </services>
    <behaviors>
      <endpointBehaviors>
        <behavior name="web">
          <webHttp/>
        </behavior>
      </endpointBehaviors>
    </behaviors>
  </system.serviceModel>
  <system.web>
    <compilation debug="true" targetFramework="4.0"/>
    <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/>
  </system.web>
</configuration>
```

6.2 Realizacija Android aplikacije

Android aplikacija je realizovana u programskom okruženju Eclipse (Helios) uz pomoć programskog jezika Java.

Projekat je podeljen na nekoliko paketa prema tome koju vrstu klasa sadrži.



Slika 11. Struktura projekta

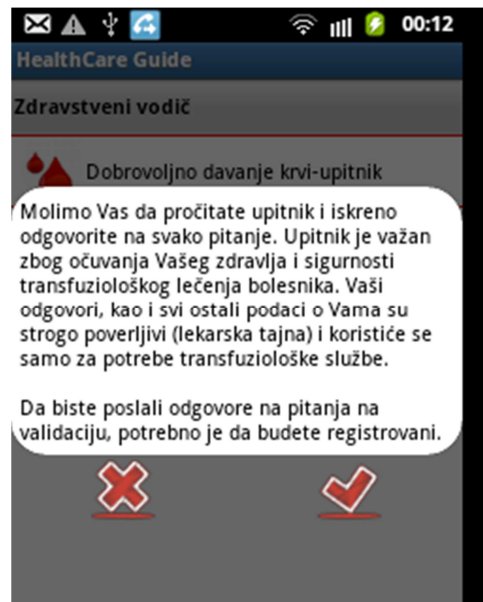
U prvom paketu nalaze se klase, tzv. Activity, koji predstavljaju ekrane koji se prikazuju korisniku aplikacije.

Nakon što se instalira aplikacija korisniku se prikaže ekran sa raznim opcijama:



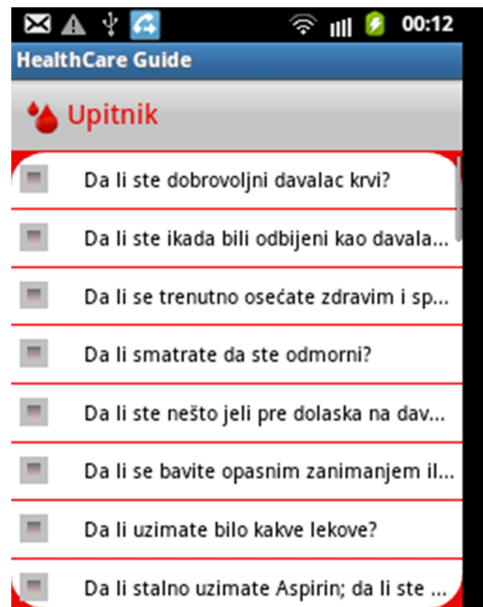
Slika 12. Početni ekran Android aplikacije

Izborom na prvu stavku – Dobrovoljno davanje krvi, otvoriće se prozor sa važnim obaveštenjima za korisnika.



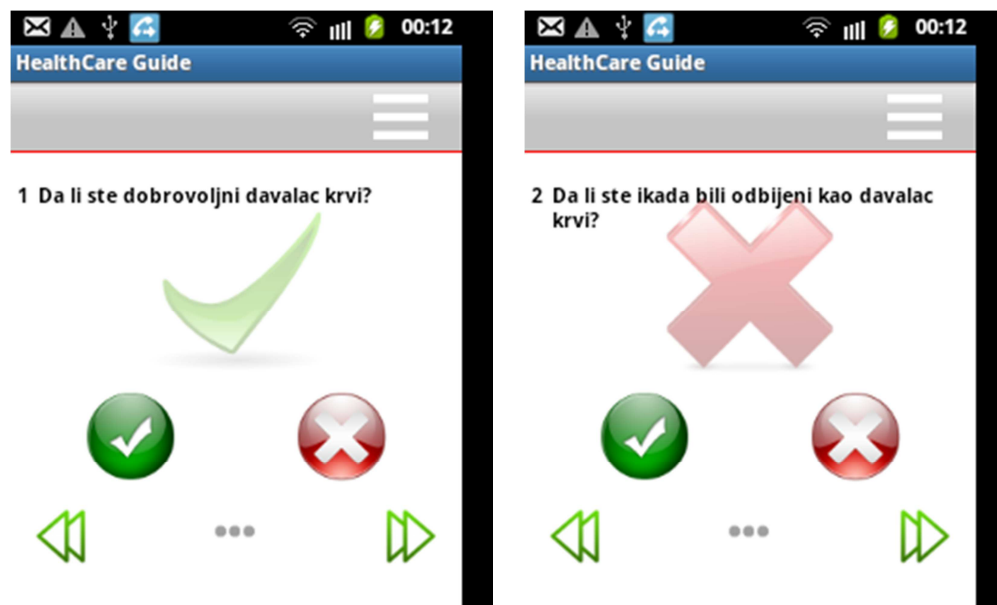
Slika 13. Obaveštenje za korisnika nakon što izabere stavku 'Dobrovoljno davanje krvi-upitnik'

Nakon što prihvati date mu sugestije prikazaće se ekran sa listom pitanja:



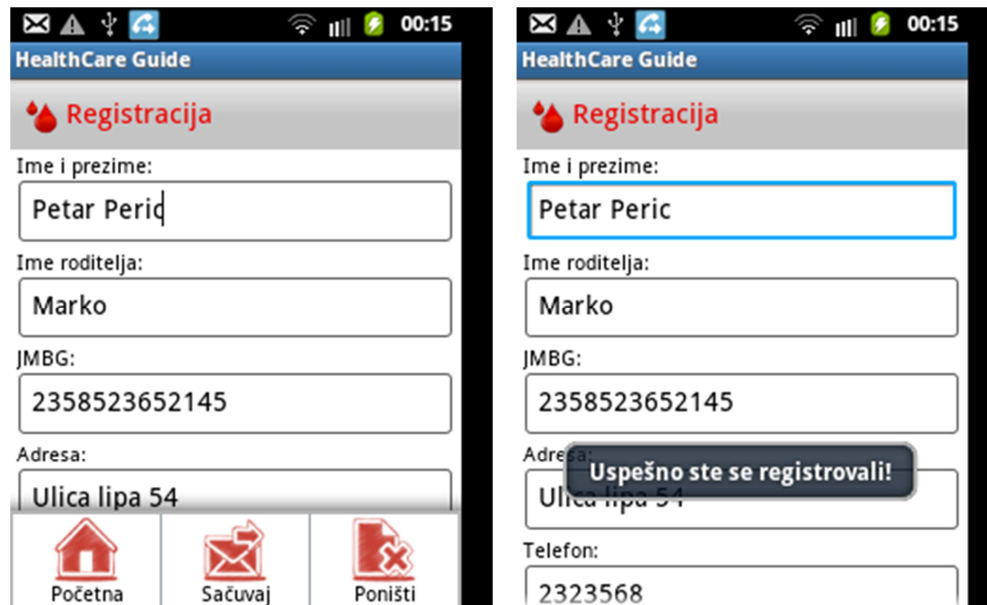
Slika 14. Lista pitanja

Izborom svakog pojedinačnog pitanja ulazi se u ekran gde je moguće dati odgovor na pitanje klikom na odgovarajuću ikonu. Na ovim ekranima implementirana je opcija Swipe, pa je moguće kretati se kroz pitanja i prelaskom prstom preko ekrana udesno ili ulevo.



Slika 15. i 16. Ekran koji prikazuje detalje izabranog pitanja, sa mogućim potvrdnim i odričnim odgovorom

Korisnik ima opciju da se registruje kako bi njegovi podaci i podaci o odgovorima na pitanja bili sačuvani na serveru.



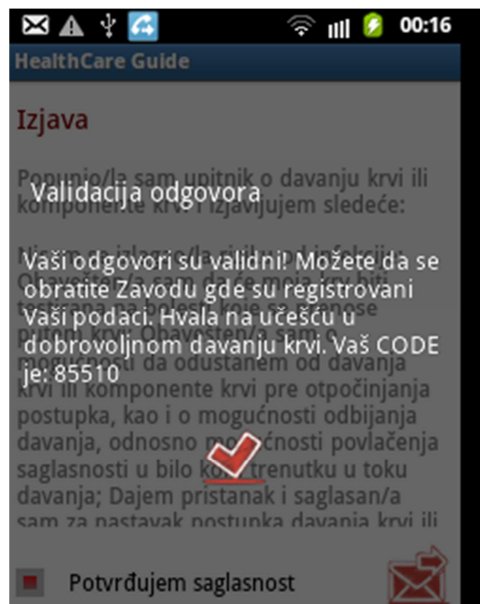
Slika 17. i 18. Ekрани za registraciju korisnika, pre registracije (17) i sa porukom o uspešnoj registraciji (18)

Tek nakon uspešne registracije moguće je proslediti odgovore pitanja serveru kako bi se izvršila validacija datih odgovora.

Ukoliko su svi odgovori zadovoljavajući korisniku se šalje poruka o tome, zajedno sa kodom. Ukoliko odgovori nisu zadovoljavajući, takođe se šalje poruka uz zahvalnost na iskrenim odgovorima. Pre slanja pitanja prikazana je izjava koja se potvrđuje klikom na polje za potvrdu.



Slika 19. i 20. Ekranu gde je korisniku omogućeno da pošalje svoje odgovore do servera (19) i ekran sa porukom o izjavi pre slanja odgovora (20)



Slika 21. Potvrda o validnosti odgovora sa kodom

U prilogu ovog rada nalaze se originalni upitnici i izjave.

Drugi paket sadrži razne vrste klasa koje se bave operacijama koje nisu vidljive korisniku, a neophodne su za normalno funkcionisanje aplikacije. To su npr. klasa *MemoryCache.java* koja je realizovana po Singleton paternu, a služi za čuvanje podataka u toku rada aplikacije. Tu se čuva lista pitanja, trenutni odgovori na pitanja i sl. *ServerCall.java* je veoma važna klasa koja rukovodi pozivima servisa.

Klasa *ServerCall.java* i metod *callWcfGetQuestionService ()* koji poziva servis za listu pitanja.

```
package singi.master.backend;
...
public class ServerCall {

    private static final String TAG = ServerCall.class.getName();
    private static ServerCall instance = null;
    public static ServerCall getInstance() {
        ....
    }
    ...
}

public String callWcfGetQuestionService() {
    DefaultHttpClient httpClient = new DefaultHttpClient();
    HttpGet request = new HttpGet(Constants.URL_WCF_QUESTIONS +
"/data");
    request.setHeader("Content-type", "application/json");
    String result = "";
    HttpResponse response;
    try {
        response = httpClient.execute(request);
        HttpEntity responseEntity = response.getEntity();

        InputStream stream = responseEntity.getContent();
        result = convertStreamToString(stream);
        stream.close();
    } catch (IllegalStateException e) {
        MyLog.e(TAG, "error on callWcfGetQuestionService(): "
            + e.getMessage().toString());
        result = Constants.ERROR_NO_WS;
    } catch (IOException e) {
        MyLog.e(TAG, "error on callWcfGetQuestionService(): "
            + e.getMessage().toString());
        result = Constants.ERROR_NO_WS;
    }
    return result;
}
```

Sam poziv servisa odvija se u zasebnoj pozadinskoj niti, tzv. *AsyncTask <Params, Progress, Result>*

Klasa *QuestionsAsyncTask* ()

```
public class QuestionsAsyncTask extends AsyncTask<Void, Integer, Void> {

    private static final String TAG = QuestionsAsyncTask.class.getName();

    private QuestionsActivity activity;
    private String result;
    private boolean error;

    public QuestionsAsyncTask(QuestionsActivity activity) {
        this.activity = activity;
    }

    @Override
    protected Void doInBackground(Void... params) {
        MyLog.d(TAG, "doInBackground()");
        result = ServerCall.getInstance()
            .callWcfGetQuestionService();
        if (result.equals(Constants.ERROR_NO_WS))
            error = true;
        return null;
    }

    @Override
    protected void onPostExecute(Void res) {
        MyLog.d(TAG, "On post Execute ()");
        activity.setToMemoryQuestions(result, error);
    }
}
```

U trećem paketu smeštene su modelske klase: *Question.java* i *Registration.java* kao i dva interfejsa *IFinished.java* i *ISwipe.java*.

U četvrtom paketu je smeštena klasa *DetailsGestureListener.java* koja osluškuje pokret Swipe.

Peti paket sadrži pomoćne klase, kao što su klasa sa konstantama, Util klasa, klasa u kojoj se obavlja parsiranje JSON-a.

Klasa *QuestionJSONParser.java*

```

public class QuestionJSONParser {
    private static final String TAG = QuestionJSONParser.class.getName();
    public static ArrayList<Question> getQuestionsArray(String js) {
        ArrayList<Question> questions = new ArrayList<Question>();
        JSONArray jsonArray = null;
        JSONObject jobject;
        try {
            jobject = new JSONObject(js);
            jsonArray = jobject.getJSONArray("GetQuestionsResult");

        } catch (JSONException e1) {
            MyLog.e(TAG, "Error: " + e1.toString());
        }
        int size = jsonArray.length();
        for (int i = 0; i < size; i++) {
            JSONObject json;
            try {
                json = jsonArray.getJSONObject(i);
                Question q = new Question();
                q.setId(json.getInt("id"));
                q.setQuestion(json.getString("question"));
                q.setAnswer(json.getBoolean("answer"));
                questions.add(q);
            } catch (JSONException e) {
                MyLog.e(TAG, "Error: " + e.toString());
            }
        }
        return questions;
    }
}

```

U šestom paketu se nalaze pomoćne klase za klase iz prvog paketa, i tu su smeštene promenljive iz 'layout-a'. Layout je XML fajl u kome se definiše izgled ekrana i izgleda ovako:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/DARK_RED"
    android:orientation="vertical" >
    <include
        android:id="@+id/header_blocks"
        layout="@layout/header_block" />
    <TextView
        android:id="@+id/tvMessage"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@drawable/rounded_list"
        android:gravity="center"
        android:text="@string/list_message_loading"
        android:textColor="@color/BLACK"
        android:textSize="17dp" />
    <include layout="@layout/questions_list" />
</LinearLayout>

```

7 Zaključak

Primena informacionih i telekomunikacionih tehnologija u medicini unapređuje efikasnost, rezultate i kvalitet medicinskih i poslovnih procesa koje sprovode odgovarajuće ustanove, stručno osoblje, pacijenti i osiguravajuća društva sa ciljem da se poboljša zdravstveno stanje pacijenta.

Razvijene zemlje sveta, zahvaljujući dobro organizovanim preventivnim programima u zdravstvenom sistemu ostvaruju uštede u lečenju koje se mere milijardama dolara. Prevencija podrazumeva aktivnosti koje se sprovode da bi se bolest sprečila, a ukoliko ipak dođe do bolesti, da se otkrije u ranoj fazi radi što uspešnijeg lečenja.

U poslednjih nekoliko godina zabeležen je ogroman porast broja mobilnih uređaja kod stanovništva zemalja u razvoju; čak 2,2 milijarde mobilnih telefona je u upotrebi u zemljama u razvoju. U zemljama u razvoju, tako i u našoj zemlji, primena mHealth bi omogućila bolju informisanost u vezi za zdravljem, sprovođenje preventivnih programa i edukaciju stanovništva.

Ovakve vrste aplikacija su značajne jer predstavljaju alat kojim se jako brzo dolazi do cilja kod korisnika pametnih telefona, kako u razvijenim zemljama tako i u zemljama u razvoju.

U Srbiji je donet program za period 2009. do 2015. godine čija će primena omogućiti nesmetano i kvalitetno funkcionisanje svih delova sistema zaštite.

U našoj zemlji nisu uočeni slučajevi implementacije mHealth sistema ili korišćenja mobilnih uređaja u svrhu pružanja zdravstvenih usluga ili informacija, osim ako se ne radi o SMS uslugama u vidu pružanja zdravstvenih informacija ili podsetnika. Pacijenti, odnosno korisnici zdravstvenih usluga i dalje zdravstvene informacije, usluge i savete primaju isključivo u ustanovama za pružanje zdravstvene zaštite.

Mnogobrojne postojeće aplikacije u svetu pokazuju da korišćenje mobilne infrastrukture u oblasti zdravlja značajno može doprineti širenju informacija i promociji zdravlja, a potencijali za dalji razvoj su ogromni.

Mnogi projekti sprovedeni u zemljama u razvoju ističu konkretne benefite nakon primene inovacija iz ove oblasti: povećan je pristup zdravstvenim uslugama i informacijama, pogotovo nedostižnim delovima populacije, poboljšana je mogućnost za dijagnostiku i praćenje bolesti, dobijaju se pravovremene informacije iz oblasti javnog zdravlja, omogućen je lakši pristup savremenoj medicinskoj edukaciji i obuci za zdravstvene radnike.

Aplikacija koju smo realizovali, vodič za dobrovoljno davanje krvi, jeste primer kako je moguće putem mobilnog telefona približiti važne informacije korisniku zdravstvenih usluga, a od korisnika dobiti povratnu informaciju. Aplikacija je urađena na osnovu originalnog upitnika koji popunjavaju eventualni dobrovoljni davaoci krvi. Osim primera kojeg smo realizovali, moguće je implementirati brojne druge forme

kojima bi se vršilo epidemiološko praćenje zdravstvenog stanja (upitnik o zaraznim bolestima, upitnik o rizicima na HIV i sl.), smanjili redovi čekanja u zdravstvenim ustanovama (upitnik za sistematske preglede, upitnik za vozače i sl.), izvršili monitoring i evaluacija (razni upitnici o kvalitetu zdravstvenih usluga).

8 Prilozi

8.1 Upitnik za davaoca – lični podaci o davaocu

  	СЛУЖБА ТРАНСФУЗИЈЕ КРВИ СРБИЈЕ УПИТНИК ЗА ДАВАОЦА (ову страну попуњава службено лице)
---	---

Шиф. даваоца: _ _ _ _ _ _ _ _		Крвна група: _____ Фенотип: _____	
Име и презиме: _____ Име родитеља: _____			
ЈМБГ: _ _ _ _ _ _ _ _ _ _ _ _ _ _		Датум рођења: _____ Пол: _	
Место становања: _____		Адреса: _____ Општина: _____	
Предузеће, факултет, школа: _____		Занимање: _____	
Тел. кућа _____ тел. посао _____ мобилни тел. _____			
Датум давања: _ _ _ _ _ _ _ Бр. претходних давања: ____ Наменско давање: Да Не			
Тип давања: Стандардно _ Аферезно _			
Регистрација и пријем даваоца		Бар код лара	
Потпис _____			
Одређивање хемоглобина/хематокрита		Одређивање крвне групе на плочици	
Бакар сулфат _ Хемоглобинметар _		А Б АВ О	
Нормалан _ Очитана вредност _____		Потпис _____	
Низак _			
Лекарски преглед		Давалац	
плућа _____ срце _____ ТА _____ / _____		Прихваћен	
ТТ _____ ТВ _____		Одбијен	
тип кесе _____		Разлог одбијања _____	
Напомена: _____		Потпис лекара _____	
Припрема кесе		Број лота кесе	
Потпис _____			
Венеункција			
<u>место пункције</u>		<u>количина узете крви</u>	
- лева рука _		405-495ml _	
- десна рука _		< 405 ml _	
		> 495 ml _	
Почетак давања: _____ сати _____ мин		Завршетак давања: _____ сати _____ мин	
Разлог привременог прекида давања _____			
Потпис: _____			

8.2 Obrazac sa pitanjima i saglasnost

ПОПУЊАВА ДАВАЛАЦ		
Молимо Вас да прочитате упитник и искрено одговорите на свако питање. Упитник је важан због очувања Вашег здравља и сигурности трансфузиолошког лечења болесника. Ваши одговори, као и сви остали подаци о Вама су потпуно поверљиви (лекарска тајна) и користеће се само за потребе трансфузиолошке службе.		
1.	Да ли сте добровољни давалац крви?	ДА НЕ
2.	Да ли сте икада били одбијени као давалац крви?	ДА НЕ
3.	Да ли се тренутно осећате здравим и способним да дате крв?	ДА НЕ
4.	Да ли сматрате да сте одморни?	ДА НЕ
5.	Да ли сте нешто јели пре доласка на давање крви?	ДА НЕ
6.	Да ли се бавите опасним занимањем или хобијем?	ДА НЕ
7.	Да ли узимате било какве лекове?	ДА НЕ
8.	Да ли стално узимате Аспирин; да ли сте га узимали у последњих пет дана ?	ДА НЕ
9.	Да ли сте до сада испитивани или лечени у болници?	ДА НЕ
10.	Да ли сте тренутно на медицинском испитивању или боловању?	ДА НЕ
11.	Да ли сте вадили зуб у протеклих 7 дана?	ДА НЕ
12.	Да ли сте у последњих 7 до 10 дана имали температуру преко 38, кијавицу, прехладу или узимали антибиотике?	ДА НЕ
13.	Да ли сте примили било коју вакцину или серум у протеклих 12 месеци?	ДА НЕ
14.	Да ли сте у последњих 6 месеци нагло изгубили на тежини без држања дијете?	ДА НЕ
15.	Да ли сте имали убоде крпеља у протеклих 12 месеци и да ли сте се због тога јављали лекару?	ДА НЕ
16.	Да ли сте икада лечени од падавице, шећерне болести, астме, туберкулозе, инфаркта, можданог удара, малигних обољења или маларије?	ДА НЕ
17.	Да ли болујете од неке друге хроничне болести: срца, плућа, бубрега, јетре, желуца и црева, костију и зглобова, нервног система, крви и крвних судова?	ДА НЕ
18.	Да ли сте икада боловали од штитасте жлезде или хипофизе и/или примали хормоне?	ДА НЕ
19.	Да ли имате неке промене на кожи или болујете од алергије?	ДА НЕ
20.	Да ли дуго крварите после повреде или спонтано добијате модрице?	ДА НЕ
21.	Да ли сте у протеклих 6 месеци:	
	а) имали неку операцију или примили крв?	ДА НЕ
	б) путовали или живели у иностранству?	ДА НЕ
	в) имали акупунктуру, пирсинг или тетоважу?	ДА НЕ
22.	Да ли сте пили алкохол у последњих 6 сати?	ДА НЕ
23.	Облици ризичних стања и понашања:	
	а) Да ли сте боловали или болујете од хепатитиса Б или Ц?	ДА НЕ
	б) Да ли мислите да је постојала могућност да се заразите ХИВ-ом?	ДА НЕ
	в) Да ли сте икада користили интравенске дроге?	ДА НЕ
	г) Да ли сте икада користили препарате за боди билдинг (стероиде)?	ДА НЕ
	д) Да ли сте икада за пружање сексуалних услуга узимали новац или дрогу?	ДА НЕ
24.	Да ли сте имали сексуалне односе током протеклих 6 месеци:	
	а) са особом која је ХИВ позитивна?	ДА НЕ
	б) са особом која има хепатитис Б или Ц?	ДА НЕ
	в) са особом која је икада за пружање сексуалних услуга узимала новац или дрогу?	ДА НЕ
	г) са особом која је икада користила интравенске дроге?	ДА НЕ
	д) са особом чије Вам дотадашње сексуално понашање није познато?	ДА НЕ
	ђ) да ли сте Ви имали аналне сексуалне односе током протеклих 6 месеци?	ДА НЕ
За жене		
25.	Да ли сте у другом стању?	ДА НЕ
26.	Да ли тренутно имате менструацију?	ДА НЕ
27.	Да ли сте имали прекид трудноће у последњих 6 месеци?	ДА НЕ
28.	Да ли сте у последњих 6 месеци имали порођај?	ДА НЕ
САГЛАСНОСТ ДАВАОЦА		
Попунио/ла сам упитник о давању крви или компоненте крви и изјављујем следеће:		
Нисам се излагао/ла ризику од инфекције		
Обавештен/а сам да ће моја крв бити тестирана на болести које се преносе путем крви		
Обавештен/а сам о могућности да одустанем од давања крви или компоненте крви пре отпочињања поступка, као и о могућности одбијања давања, односно могућности повлачења сагласности у било ком тренутку у току давања		
Дајем пристанак и сагласан/а сам за наставак поступка давања крви или компонената крви		
Обавештен/а сам о сврси давања крви или компонената крви		
Упознат/а сам са уобичајеним ризицима и могућим реакцијама током узимања крви, као и о обиму тестирања крви		
Обавештен/а сам о заштити података о личности		
Имао/ла сам могућност да постављам питања		
Добио/ла сам задовољавајуће одговоре на сва постављена питања		
Део/ла сам истините информације према свом најбољем знању и потврђујем веродостојност датих података		
ХВАЛА ШТО ДАЈЕТЕ КРВ		
Потпис даваоца _____		
Напомена:		
◆ Одустао/ла		
◆ Одбијање давања		
◆ Сагласност повучена		
Потпис даваоца _____		

9 Literatura

1. World Health Organization. Available: <http://www.who.int/trade/glossary/story021/en/index.html>, mart, 2012.
2. G. Eysenbach. What is e-health?, J Med Internet Res, 2001;3(2):e20
3. Ministarstvo nauke i životne sredine Republike Srbije. Strategija razvoja informacionog društva u Republici Srbiji. Službeni glasnik RS, br. 55/05 i 71/05 ispravka
4. Kuntagod N., Mukherjee C. Mobile decision support system for outreach health worker. Healthcom, Proc. 2011 13th IEEE International Conference on e-Health Networking Applications and Services, pp. 56-59, 13-15 June 2011.
5. Nkosi M.T., Mekuria F., Gejibo S.H. Challenges in mobile bio-sensor based mHealth development. Healthcom, Proc. 2011 13th IEEE International Conference on e-Health Networking Applications and Services, pp. 21-27, 13-15 June 2011.
6. Vital Wave Consulting. mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World. Washington, D.C. and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership, 2009.
7. Ministarstvo zdravlja. Razvoj zdravstvenog informacionog sistema za osnovne zdravstvene i farmaceutske usluge u Srbiji, Euro Health Group, 2005, rev. VI 06.07.05.
8. Uredba o programu rada, razvoja i organizacije integrisanog zdravstvenog informacionog sistema –“E-zdravlje”. Sl. glasnik RS, br. 55/2009.
9. Draft Guidance for Industry and Food and Drug Administration Staff-Mobile Medical applications. July, 2011. Available: <http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM263366.pdf>, july, 2012.
10. mHealth insight. So is a mhealth record just a “Mobile Optimised” Electronic health record? Available: <http://mhealthinsight.com/2010/01/11/so-is-a-mhealth-record-just-a-mobile-optimised-electronic-health-record/>, maj, 2012.
11. Story from BBC news. Available: http://news.bbc.co.uk/2/hi/programmes/click_online/8639590.stm, mart, 2012.
12. Gartner. Available: <http://www.gartner.com/it/page.jsp?id=1924314>, april, 2012.
13. Canalys. Smart phones overtake client PCs in 2011. Available: <http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011>, april, 2012.
14. The World's First Smartphone – IBM's Simon. Available: <http://bb-iphone-zone.com/2011/the-worlds-first-smartphone-ibms-simon/>, maj, 2011.
15. Bluetooth. More than 40 million devices and counting. Available: <http://www.bluetooth.com/pages/medical.aspx>, maj, 2011.
16. World Health Organization Global Observatory for eHealth. mHealth: New horizons for health through mobile technologies: second global survey on eHealth. Global observatory for eHealth series. Volume 3. WHO. 2011.
17. Mobile Communications for Medical Care: a study of current and future healthcare and health promotion applications, and their use in China and elsewhere. Final report. University of Cambridge, China Mobile. 2011.

18. Wikipedia. iOS. Available: <http://en.wikipedia.org/wiki/IOS>, mart, 2012.
19. Wikipedia. Windows Mobile. Available: http://en.wikipedia.org/wiki/Windows_Mobile, mart, 2012.
20. Wikipedia. Windows Phone. Available: http://en.wikipedia.org/wiki/Windows_Phone, mart, 2012.
21. Wikipedia. Symbian. Available: <http://en.wikipedia.org/wiki/Symbian>, mart, 2012.
22. Wikipedia. BlackBerry. Available: <http://en.wikipedia.org/wiki/BlackBerry>, mart, 2012.
23. Royal Tropical Institute. What is mHealth? Available: <http://www.mhealthinfo.org/what-mhealth>, juli, 2012.
24. Europe's Information Society. ICT for better healthcare in Europe. Available at: http://ec.europa.eu/information_society/activities/health/index_en.htm, jun, 2012.
25. Covic, Z., et al. Development of integrated model for health promotion in medical informatics. Information Technology Interfaces (ITI), Proc. 2011 IEEE 13th International Conference on e-Health Networking, Applications and Services, pp.371-378, 27-30 June 2011.
26. Vatsalan D., et al. Mobile Technologies for Enhancing eHealth Solutions in Developing Countries. eTELEMED, Proc. 2010 Second International Conference on eHealth, telemedicine, and Social Medicine, pp.84-89, 10-16 February 2010.
27. Hunter D., et al. Od početka... XML, CET, Beograd, 2000.
28. ASP.NET sa C#, A. Russell Jones, Kompjuter biblioteka, Čačak, 2003.
29. Introducing JSON. Available: <http://www.json.org/>, jun, 2012.
30. Kreger H. Web Services Conceptual Architecture (WSCA 1.0). IBM Software Group, 2001.
31. Fielding T. Architectural Styles and the Design of Network-based Software Architectures. Dissertation, University of California, Irvine, 2000.
32. Thomas J.P. Modeling of Web services flow. E-Commerce, Proc. 2003. IEEE International Conference on E-Commerce, pp. 391-398, 24-27 June 2003.
33. Microsoft. Razvoj XML Web servisa i serverskih komponenti sa MS VB.NET i MS VC#.NET, CET, Beograd, 2003.
34. Saravanakumar's WCF tutorial. Available: <http://wcftutorial.net>, jun, 2012.
35. Chappell D. Introducing Windows Communication Foundation. Microsoft, 2010.
36. Vawter C., Roman E. J2EE vs. Microsoft.NET: A comparison of building XML-based web services. The ServerSide.com. Available: <http://www.theserverside.com>, juli, 2012.
37. Price J., Gunderloy M. Visual C#.NET, Kompjuter biblioteka, Čačak, 2003.
38. Vawter C., Roman E. J2EE vs. Microsoft.NET: A comparison of building XML-based web services. The ServerSide.com. Available: <http://www.theserverside.com>, juli, 2012.
39. Chung F. Custom Class Loading in Dalvik. Google. Available: <http://android-developers.blogspot.com>, juli, 2012.
40. Conder S., Darcey L. Learn Java for Android Development: Introduction to Java. Available: <http://mobile.tutsplus.com>, juli, 2012.
41. Aniszczyk C., Gallardo D. Get started with the Eclipse Platform. DeveloperWorks. Available: <http://www.ibm.com>, juli, 2012.