

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Boštjan Vukič

IVR kot orodje telemedicine

Diplomsko delo

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Boštjan Vukič

Mentorica: prof. dr. Valentina Hlebec

IVR kot orodje telemedicine

Diplomsko delo

Ljubljana, 2017

ZAHVALA za možnost študija in diplomsko delo gre podjetju Telekom Slovenije, d.d., ki mi je omogočilo študij, in vsem sodelavcem ter nadrejenim, ki so mi v vseh letih študija stali ob strani, me spodbujali in mi hkrati tudi zaupali, da bom to nalogo izpeljal do konca.

Največja zahvala pa gre mojim bližnjim, predvsem ženi Simoni, hčerki Kim, mami Zdenki in očetu Mladenu. Iskreno se vam zahvaljujem za ljubezen, potrpljenje in podporo. Hvala, ker ste verjeli vame in mi v ključnih trenutkih pomagali, da sem premagoval oviro za oviro in sem danes prispel do cilja.

In nenazadnje se zahvaljujem tudi sošolcema Milošu in Klemnu. Hvala vama za izjemno kolegialnost in podporo. V vajini družbi je čas študija minil hitro in nepozabno.

IVR kot orodje telemedicine

Hiter razvoj internetnih storitev vstopa na področje medicine in zdravja. Tudi javni zdravstveni sistem se spreminja in uvaja rešitve, ki uporabnikom interneta omogočajo lažji in preglednejši dostop do informacij in zdravstvenih storitev. A podatki Statističnega Urada Republike Slovenije kažejo, da kar dobra petina oseb v Sloveniji ne uporablja interneta. Med njimi sta kar dobri dve tretjini oseb starejši od 65 let. Tem osebam so nove storitve in informacije nedostopne. V diplomskem delu sem zasnoval in izdelal aplikacijo telezdravje.si, ki integrira spletni in glasovni uporabniški vmesnik. Z glasovnim uporabniškim vmesnikom je aplikacija dostopna prek javnega telefonskega omrežja tudi uporabnikom, ki ne uporabljajo interneta. Z uporabo glasovnega uporabniškega vmesnika lahko uporabniki vnašajo različne tipe podatkov, kot so besede, cela in decimalna števila. Vnos poteka z uporabo govora ali tipk telefona. Na ta način lahko uporabniki spremljajo potek svojih kroničnih bolezni. Aplikacija uporablja odprte standarde, kot so VoiceXML, Speech Recognition Grammar in HTML5. Izdelana je v programskem jeziku Java in se izvaja na zanesljivih ter preverjenih komponentah, ki v principu demonstrirajo uporabo izbranih tehnologij, ki so primerne tudi za večje sisteme.

Ključne besede: IVR, telemedicina, telezdravje, telekomunikacije.

IVR as a telemedicine tool

Fast development of internet services is entering into medical and health segments. Even public health system is transforming and implementing new solutions which provide internet users with easy and organized access to information and health services. However, data from Statistical Office of the Republic of Slovenia are indicating that more than one fifth of people in Slovenia are not using the internet. And among them there are more than two thirds of people older than 65 years. For these people new services and information are inaccessible. In the thesis I have designed and implemented the telezdravje.si application which integrates a web and a voice user interface. By using voice user interface over the public telephone network the application is available even for those users who do not use the internet. With the voice user interface users can input different types of data such as: words, integer and decimal numbers. Input is done by using speech or phone keys. In this way users can keep track of their chronic medical conditions. The application is using open standards as for instance VoiceXML, Speech Recognition Grammar and HTML5. It is developed in Java programming language and is running on verified and stable components that showcase the use of chosen technologies, which are suitable even for larger systems.

Keywords: IVR, telemedicine, telehealth, telecommunications.

KAZALO VSEBINE

SEZNAM KRATIC	7
1 UVOD	9
2 CILJ IN PREDMET DIPLOMSKEGA DELA.....	11
3 METODOLOGIJA	12
4 OPREDELITEV POJMOV IN UPORABLJENIH TEHNOLOGIJ	13
4.1 Telemedicina, telezdravje, e- medicina in e-zdravje	13
4.2 IVR	14
4.2.1 VoiceXML.....	15
4.2.2 Speech Recognition Grammar.....	16
4.2.3 HP OpenCall Media Platform	16
4.2.4 Scansoft Open Speech Recognizer	17
4.3 Java	17
4.3.1 Java Servlet.....	17
4.3.2 Knjižnica Spring Framework	18
4.3.3 Knjižnica Hibernate ORM.....	18
4.3.4 Knjižnica Thymeleaf	18
4.4 Apache Tomcat.....	18
4.5 MySQL Community Server.....	19
4.6 HTML5, CSS in JavaScript	19
4.6.1 Knjižnica jQuery	20
4.6.2 Knjižnica Bootstrap	21
4.6.3 Knjižnica Chartist.js	21
5 IZBIRA BOLEZNI IN OPREDELITEV ZDRAVSTVENIH PARAMETROV.....	22
6 NAČRTOVANJE, IZDELAVA IN PREDSTAVITEV APLIKACIJE.....	25
6.1 Načrtovanje aplikacije	25
6.2 Izdelava aplikacije	29
6.2.1 Konceptualna zasnova	29
6.2.2 Podatkovni model.....	30
6.3 Programiranje spletnega uporabniškega vmesnika.....	32
6.3.1 Programiranje glasovnega uporabniškega vmesnika.....	33
6.4 Predstavitev aplikacije	34
6.4.1 Spletni uporabniški vmesnik	34
6.4.2 Glasovni uporabniški vmesnik	36

7	SKLEP.....	38
8	LITERATURA.....	40
	PRILOGE.....	43
	Priloga A: Izvorna koda projekta telezdravje.si.....	43
	Priloga B: Speech Recognition Grammar.....	43
	Priloga C: Call Data Record.....	47

KAZALO TABEL

	Tabela 5.1: Parametri za spremljanje bolezni.....	24
	Tabela 6.1: Preslikava entitetnih tipov v tabele baze podatkov.....	30
	Tabela 6.2: Vprašanja v glasovnem vmesniku.....	37

KAZALO SLIK

	Slika 6.1: Proces registracije.....	26
	Slika 6.2: Proces prijave – splet.....	26
	Slika 6.3: Proces prijave – telefon.....	26
	Slika 6.4: Proces vnosa podatkov.....	27
	Slika 6.5: Proces izpisa podatkov.....	27
	Slika 6.6: ER-diagram baze podatkov.....	28
	Slika 6.7: Konceptualna zasnova rešitve telezdravje.si.....	30
	Slika 6.8: Fizična shema baze podatkov.....	32
	Slika 6.9: Prijava in registracija.....	35
	Slika 6.10: Vnos in namizje.....	35
	Slika 6.11: Zgodovina.....	36

SEZNAM KRATIC

API	(angl. <i>application programming interface</i>) aplikacijski programski vmesnik
ASR	(angl. <i>automated speech recognition</i>) funkcija samodejne prepozne govora
CALM	(angl. <i>collaborative application lifecycle management</i>) skupinsko upravljanje življenjskega cikla aplikacije
CATI	(angl. <i>computer-assisted telephone interviewing</i>) računalniško podprto telefonsko anketiranje
CCXML	(angl. <i>call control XML</i>) XML za krmiljenje poziva
CSS	(angl. <i>cascading style sheets</i>) predloge, ki določajo videz spletnih strani
DBMS	(angl. <i>database management system</i>) sistem za upravljanje baz podatkov
DOM	(angl. <i>document object model</i>) objektni model dokumenta
DTMF	(angl. <i>dual-tone multi-frequency signaling</i>) telefonska signalizacija z uporabo sinteze dveh tonov v slišnem spektru
ER	(angl. <i>entity-relationship</i>) entiteta-relacija
HP OCMF	(angl. <i>HP OpenCall Media Platform</i>) produkt podjetja HP, ki omogoča IVR funkcije
HQL	(angl. <i>Hibernate Query Language</i>) programski jezik, ki ga uporablja knjižnica Hibernate za tvorjenje SQL-poizvedb
HTML	(angl. <i>hypertext markup language</i>) hiperbesedni označevalni jezik, ki se uporablja za izdelavo spletnih strani in ga prikazujejo spletni brskalniki
HTML5	HTML, različica 5
HTTP	(angl. <i>hypertext transfer protocol</i>) aplikativni protokol za komunikacijo med odjemalcem in strežnikom v IP-omrežjih
IP	(angl. <i>internet protocol</i>) protokol za usmerjanje podatkovnih paketov v paketnih omrežjih, kot je internet
IVR	(angl. <i>interactive voice response</i>) interaktivni glasovni odzivnik
IVVR	(angl. <i>interactive voice and video response</i>) interaktivni glasovni in video odzivnik
Java EE	(angl. <i>Java Enterprise Edition</i>) poslovna ali razširjena različica programskega jezika Java, ki vsebuje dodatne vmesnike
JPA	(angl. <i>Java Persistence API</i>) vmesnik za preslikavo med objekti Java in

	entitetami tip v bazah podatkov
JVM	(angl. <i>Java virtual machine</i>) navidezni stroj Java ali program, ki v operacijskem sistemu izvaja binarno kodo Java
MRF	(angl. <i>media resource function</i>) komponenta za zagotavljanje medijskih funkcij v paketnem telekomunikacijskem omrežju
MSCML	(angl. <i>media server control markup language</i>) označevalni jezik za krmiljenje MRF-komponente
MVC	(angl. <i>model view controller</i>) razvojni pristop, ki se uporablja pri razvoju spletnih aplikacij, ki vsebinsko ločuje kodo v skupine: podatkovni model, pogled in krmilnik
ORM	(angl. <i>object relational mapping</i>) funkcionalnost preslikovanja objektov v relacijske baze podatkov
OSR	(angl. <i>Open Speech Recognizer</i>) produkt podjetja SscanSoft, ki omogoča ASR-funkcije
SIP	(angl. <i>session initiation protocol</i>) protokol v IP-omrežju za upravljanje s komunikacijskimi sejami
SQL	(angl. <i>structured query language</i>) programski jezik, ki se uporablja za delo z relacijskimi bazami podatkov
SURS	Statistični Urad Republike Slovenije
TTS	(angl. <i>text to speech</i>) funkcija sinteze govora iz besedila
VXML	VoiceXML
W3C	(angl. <i>world wide web consortium</i>) organizacija za standardizacijo tehnologij svetovnega spleta
WHO	(angl. <i>World Health Organisation</i>) svetovna zdravstvena organizacija
XML	(angl. <i>extensible markup language</i>) označevalni zapis dokumentov, ki se uporablja pri komunikaciji med računalniškimi sistemi in komponentami

1 UVOD

Splošno dejstvo je, da se populacija stara in da je treba administrativno delo medicinskega in zdravstvenega osebja poenostaviti. Kljub temu pa je treba ohraniti, če ne povečati, dostopnost in zanesljivost medicinskih in zdravstvenih storitev, zato je to diplomsko delo usmerjeno v raziskovanje ustreznosti in uporabnosti IKT, ki bi lahko z majhnimi investicijami in spremembami delovnih procesov optimiziral delo in izboljšal dostopnost zdravstvene oskrbe. V nalogi se zato osredotočam na možnost strojnega, telefonskega zbiranja zdravstvenih podatkov z uporabo glasovnega telefonskega odzivnika ali IVR ter moderne spletne aplikacije.

Internet je v Sloveniji v prvem četrtletju 2015 uporabljalo okoli 1.136.000 oseb ali 73 % ljudi. Od teh oseb je 83 % takih, ki internet uporabljajo vsak dan ali skoraj vsak dan. Kar 22 % oseb, starih od 16 do 74 let, pa interneta ni še nikoli uporabljalo. Teh je največ, kar 65 %, med osebami, starimi od 65 do 74 let. Sledijo osebe, stare od 55 do 64 let, 47 %, in osebe, stare od 45 do 54 let, 21 %. Zanimivo je, da kar 67 % uporabnikov interneta uporablja internet za iskanje informacij, povezanih z zdravjem. Po namenu uporabe interneta so tovrstne vsebine uvrščene kar na četrto mesto in so po uporabi višje od družabnih omrežij (Zupan 2015).

Z vidika dostopnosti spletnih vsebin in spletnih storitev je skupina oseb, starejših od 65 let, najbolj izključena. Če izpostavim samo zvišan krvni tlak, ga ima kar 76 % vseh oseb v Evropi, ki so stare od 65 do 74 let (Petek-Šter 2014, 1). To pomeni, da zbiranje zdravstvenih informacij zgolj z uporabo spletnih aplikacij, v spletnih brskalnikih, ni primerno za dobro petino Slovencev. In v tem kontekstu je za IVR-tehnologije še vedno veliko priložnosti, saj so IVR-aplikacije primerne za ljudi, ki niso računalniško pismeni (Olufemi in drugi 2009, 50). Po drugi strani pa je v Sloveniji že v zadnjem četrtletju leta 2008 penetracija uporabnikov mobilne telefonije presegla 100 % (RIS). Konec leta 2015 pa je bilo kar 2.354.000 uporabnikov mobilnega omrežja (Gostiša 2016). Torej je IVR-tehnologija dostopna praktično vsem Slovincem, ki imajo dostop do mobilnega ali fiksnega telefona.

Zbiranje podatkov z IVR-tehnologijo je kakovostno. V primerjalni študiji, ki je primerjala zbiranje podatkov z uporabo tehnologij IVR in CATI (Midanik in Greenfield 2008, 587), je bilo ugotovljeno, da na osebna in neobčutljiva vprašanja ni bilo statistično značilnih razlik med odgovori, pridobljenimi z uporabo IVR, in kontrolno skupino, ki je bila izprašana po metodi CATI. Majhne, a statistično značilne razlike, so bile opažene pri demografski skupini

moških, starih 40 let in več, pri vprašanju spolne usmeritve. Razlika je bila opažena tudi v skupini s starostjo od 18 do 39 let pri odgovoru na vprašanje o uporabi kondomov.

V IVR-aplikacijah se lahko za vnos uporablja tudi prepoznavnik govora. In prav na področju prepoznavne govora oziroma ASR se je lansko leto zaostri tehnološka tekma, ko je Google objavil aplikacijo Google Assistant (Eadicicco 2017). V produktu Google Home, ki uporablja tehnologijo Google Assistant, je na voljo napreden ASR, ki za izboljšanje natančnosti vnosa spremlja kontekst govora. Večja natančnost prepoznavne je posledica neprestanega snemanja, ki glede na predhodne prepoznavne zmanjšuje možen nabor besed v slovarju trenutne prepoznavne (Aleksic in drugi 2015, 1–2). Kontekstualno rabo ASR je že leta 2014 predstavil tudi Amazon in od takrat je na voljo v njihovem produktu Echo, sledil pa je tudi Apple z aplikacijo Siri (Eadicicco 2017). Ker je Google konec lanskega leta objavil Google Cloud Speech API, ki omogoča drugim podjetjem in programerjem, da uporabljajo njihov ASR sistem, ki trenutno podpira 80 jezikov in je hkrati tudi kontekstualen ter uporablja mehanizme strojnega učenja, da stalno izboljšuje natančnost prepoznav (Cloud Speech API), lahko sklepam, da bo v prihajajočih letih področje ASR kakovostno in v dostopnosti znatno napredovalo. Hkrati se bo spremenil tudi način uporabe elektronskih naprav, ki za vnos podatkov ne bodo več potrebovale tipk in zaslonov, kot to že vidimo pri produktih Amazon Echo in Google Home.

2 CILJ IN PREDMET DIPLOMSKEGA DELA

Cilj diplomskega dela je preučiti uporabo in delovanje IVR-aplikacij z namenom ugotavljanja, ali so telefonski govorni uporabniški vmesniki primerni za uporabo na področju telemedicine in telezdravja. Ker v Sloveniji ni takega sistema, je predmet te naloge načrtovanje in izdelava aplikacije, ki ima dva uporabniška vmesnika:

- glasovni uporabniški vmesnik za vnos podatkov – IVR,
- spletni uporabniški vmesnik za registracijo uporabnikov, vnos podatkov in urejanje podatkov.

Tako sem za potrebe raziskovanja tega področja načrtoval in izdelal aplikacijo, ki omogoča vnos osnovnih podatkov za spremljanje kroničnih bolezni z uporabo IVR-sistema in spletnega brskalnika. Ker aplikacija uporabnikom omogoča vnos podatkov prek govornega vmesnika in tudi z uporabo spletnega obrazca, sem tako lahko opravil primerjavo uporabnosti glasovnega telefonskega vmesnika s spletnim uporabniškim vmesnikom.

V nalogi torej zasledujem dve raziskovalni vprašanji:

- I. Kako načrtovati in izdelati aplikacijo za vnos in urejanje podatkov o stanju kronične bolezni, ki ponuja glasovni telefonski in spletni uporabniški vmesnik?
- II. Kakšna je uporabnost glasovnega telefonskega vmesnika v primerjavi s spletnim uporabniškim na primeru izdelane aplikacije?

3 METODOLOGIJA

Pri iskanju odgovorov na zastavljeni raziskovalni vprašanji v diplomskem delu uporabljam več različnih metod, postopkov in tehnik. Metodologija gradnje informacijskih sistemov v praksi največkrat pomeni organizacijsko-tehnično znanje, ki se uporablja pri snovanju in izdelavi računalniških sistemov in rešitev (Kovačič in Vintar 1994, 29).

Sprva sem opravil tradicionalen pregled literature, da sem lahko opredelil področje raziskovanja in opredelil pojme. Na podlagi raziskane literature sem izbral kronične bolezni in zdravstvene parametre, ki so na voljo uporabnikom aplikacije. Ta del metodologije pravzaprav sovпада s fazo razvojne metodologije CALM, ki jo je izdelalo podjetje IBM. Za razvoj aplikacije sem CALM-metodologiji sledil le po faznem sosledju. V tem kontekstu metodologija omogoča izvedbo celotnega razvoja informacijskega sistema, ki je razdeljen v več različnih faz (Kovačič in Vintar 1994, 29). V celoti pa CALM-metodologije nisem uporabljal, ker je namenjena skupinskemu delu več ekip, na več oddaljenih lokacijah, in je za moj majhen razvojni projekt neprimerna. Uporabil sem naslednje faze (IBM 2008, 56–57):

- Snovanje (angl. *inception*). V fazi snovanja sem opravil pregled literature in določil vsebino aplikacije.
- Načrtovanje (angl. *elaboration*). V fazi načrtovanja sem definiral konceptualne tehnične vsebine, kot so arhitektura, podatkovni model in procesni model.
- Implementacija (angl. *construction*). V fazi implementacije sem razvil in preizkusil aplikacijo v razvojnem okolju.
- Prehod v produkcijo (angl. *transition*). V fazi prehoda v produkcijo sem opravil teste na strežniku in v aplikaciji, ki je javno dostopna in na voljo uporabnikom.
- Produkcija (angl. *production*). Je trenutna faza, v kateri se nahaja aplikacija, in bo ostala v tej fazi, dokler jo ne bo nadomestila nova različica ali pa bo aplikacija odstranjena.

4 OPREDELITEV POJMOV IN UPORABLJENIH TEHNOLOGIJ

4.1 Telemedicina, telezdravje, e-medicina in e-zdravje

Telekomunikacijske storitve in rešitve so vstopile tudi na področje zdravstva in medicine. Stroka je v uporabo uvedla več pojmov, ki povezujejo IKT z zdravstvom in medicino. Tako pogosto zasledimo pojme: telezdravje, telemedicina, e-zdravje in e-medicina. Za moje področje raziskovanja je pomembno, da jih opredelim in definiram.

V znanstvenih revijah in strokovni literaturi se pojavlja več različnih definicij. V širšem pomenu je telemedicina kombinacija IKT, multimedije in računalniških omrežij, ki zagotavlja širok nabor zdravstvenih aplikacij in storitev (Fong in drugi 2010, 8–9). V reviji *Telemedicine and e-health* (Bashshur in Shannon 2009, 601) je navedeno, da gre za tehnologije, ki omogočajo elektronsko zajemanje, obdelavo, posredovanje, shranjevanje, pridobivanje in izmenjavo informacij za potrebe promocije zdravja, preprečevanja bolezni, zdravljenja bolnih, spremljanja kroničnih bolezni, rehabilitacijo bolnih ter varovanje javnega zdravja in zdravstvene varnosti. Sistemi telemedicine so integrirani v zdravstvena omrežja, ustanove in organizacije. Kot taki omogočajo: v bolnika usmerjeno ozdravljanje, podporo pri odločanju v klinični obravnavi, učinkovitejšo spremljanje kroničnih bolezni, promocijo zdravega načina življenja in učinkovito oskrbo bolnikov v ustanovah ter na daljavo, na domu.

Pomembno je, da izpostavimo razliko med telezdravjem in telemedicno. Telezdravje je uporaba IKT, ki omogočajo oddaljeno spremljanje in varovanje zdravja. Telemedicina pa predstavlja uporabo IKT na področju kurative. Telezdravje se tako bolj navezuje na mednarodne aktivnosti WHO na področju javnega zdravstva, telemedicina pa je usmerjena bolj v klinično uporabo (Charness in drugi 2012, 2).

Čeprav so bile prve klinične študije primerov opravljene v 70. letih (Maheu in drugi 2001, 3), si bomo pogledali nedavne primere, ki bolj relevantni za naše raziskovanje, saj temeljijo na uporabi internetnega omrežja, ki omogoča komunikacijo med sistemi, kot je na primer oddaljeno spremljanje bolnikov z uporabo IKT, kjer doktorji medicine in zdravstveno osebje oddaljeno nadzorujejo paciente. Uporaba telemedicine obljublja bolj kakovostno bivanje pacientov v domači oskrbi in skrajševanje postoperativnega bivanja v bolnici. Telemedicina pa ni le namenjena skrajševanju dobe bolnišnične oskrbe, ampak prinaša udobnejše življenje bolnikom s kroničnimi boleznimi, saj omogoča oddaljeno spremljanje bolezni in tako vpliva na kakovostnejše življenje kronično bolnih.

V Sloveniji je v letih od 2008 do 2015 na Ministrstvu za zdravje potekal projekt e-Zdravje, ki je uvedel naslednje storitve e-zdravja (Ministrstvo za zdravje):

eNaročanje zagotavlja informacijsko podporo pri napotitvi in naročanju pacientov na zdravstvene storitve s primarne zdravstvene ravni na sekundarno in terciarno ter znotraj sekundarne in terciarne. Prednosti prinaša tako pacientom kot izvajalcem zdravstvenih storitev - možnost izbire termina, pregled nad vsemi ponudniki zdravstvene storitve, učinkovito obveščanje ob morebitnih spremembah, enostavnejša komunikacija med zdravniki.

eTriaža zagotavlja informacijsko podporo za Manchesterski triažni postopek v zdravstvenih domovih ter bolnišnicah, uporablja pa se z namenom zmanjšanja kliničnega tveganja v primerih, ko zaradi velikega števila pacientov zdravstveno osebje ne zmore sprotne obravnave vseh pacientov. Namen triaže je, da zdravstveno osebje paciente strokovno razdeli glede na nujnost obravnave in jim tako zagotovi pravilno in pravočasno oskrbo. eTriaža omogoča varnejše vodenje čakalnega seznama pacientov.

TeleKap je eden izmed najuspešnejših programov projekta eZdravje, saj omogoča hitro diagnostiko in obravnavo pacientov z možgansko kapjo. V mrežo TeleKap je vključenih 12 bolnišnic po Sloveniji, ki sodelujejo z Nevrološko kliniko UKC Ljubljana, od koder zdravniki preko avdio-videokonferenčnega klica, izmenjave radioloških slik in izvidov dobijo učinkovito strokovno podporo pri oskrbi pacienta s to akutno boleznijo.

4.2 IVR

IVR ali interaktivni glasovni odzivnik je dostopen prek javnega telefonskega omrežja. Je IKT-tehnologija, ki omogoča interakcijo računalniškega sistema in uporabnika prek mobilnega ali fiksnega telefonskega omrežja. IVR se uporablja na različnih področjih, kot so na primer: prodaja vstopnic, bančništvo, kontaktni centri, televizijske oddaje in glasovna pošta. Interakcija med uporabnikom in IVR-sistemom poteka prek tipk telefona oziroma DTMF-tonov ali glasovnih ukazov z uporabo tehnologije ASR. V takem primeru je funkcija ASR tista, ki uporabniku omogoča, da z glasovnimi ukazi krmili IVR-aplikacijo in vnaša podatke v računalniški sistem. Ko so se IVR-aplikacije pojavile v začetku 90. let, so podjetja planila po novi tehnologiji, saj so obljubljale zmanjševanje stroškov dela. Prve implementacije pri uporabnikih niso bile dobro sprejete, a se je to z leti spremenilo. Danes mnogi uporabniki

predplačniških sistemov mobilne telefonije brez pomisleka uporabljajo svoj predplačniški račun z uporabo IVR-aplikacije.

IVR-aplikacije pa niso le samostojni sistemi, temveč so lahko sestavni del večjih sistemov, ki prinašajo glasovni uporabniški vmesnik. Tako je lahko nek sistem uporabnikom na voljo na več različnih dostopnih kanalih (angl. *cross-channel*) – podobno, kot velja tudi za mobilne aplikacije, ki so pogosto sestavni del večjega sistema, ki je uporabnikom dostopen prek mobilnih aplikacij ali prek spletnega brskalnika (Hura 2012).

Prednost modernih IVR-aplikacij in integracij je v tem, da omogočajo razvejano logiko, ki jo omogoča funkcija menijske izbire. Aplikacija lahko instantno shrani vnesen podatek v podatkovno skladišče, hkrati pa lahko dostopa do skladišča predposnetih odgovorov ali funkcije TTS in na podlagi vnesenih podatkov ter vgrajene logike vrne ustrezen odgovor. Omogočajo tudi snemanje sporočil, odhodno klicanje in prevezovanje klicev. IVR je med drugim primeren tudi za uporabo v kliničnih študijah za spremljanje kroničnih bolezni, na primer: zdravljenje kronične bolečine (Lieberman in Naylor 2012), raziskovanje tveganj za nastanek raka dojke (Llanos in drugi 2014) in za nevropsihološko testiranje (Mundt in drugi 2006).

Telekom Slovenije v skupini produktov M:Vrata ponuja dostop do vmesnika za izdelavo klicnega portala, ki temelji na tehnologiji VoiceXML. Med ključnimi lastnostmi sta izpostavljeni: več sočasnih telefonskih zvez in razpoložljivost 24 ur na dan ter 7 dni v tednu. Nekateri primeri uporabe, ki jih navajajo, so: vedeževanja, horoskopi, zasebni stiki, valutni tečaji, stanje v prometu, snežne razmere, nasveti, loterijski in športni rezultati, nagradne igre, ankete, glasovanja (Telekom Slovenije).

4.2.1 VoiceXML

VoiceXML je standardiziran format za programiranje glasovnih brskalnikov, ki omogočajo glasovni uporabniški vmesnik prek telefonskega dostopnega kanala. Je odprt standardiziran format zapisa, ki ga je standardizirala organizacija W3C. VoiceXML-forum so ustanovila podjetja AT&T, IBM, Lucent in Motorola. Ustanovljen je bil z namenom, da definira standarden programski jezik, ki mora biti zasnovan na dialogu in programerjem omogoča razvoj pogovornih aplikacij. Za zapis so izbrali XML-format zapisa, saj so takratni tehnološki trendi kazali, da bo v prihodnosti XML-format igral ključno vlogo pri izmenjavi podatkov med informacijskimi sistemi. Prva različica (1.0) je bila javno objavljena leta 2000. Leta 2004 ji je sledila različica 2.0, ki jo je leta 2006 nadgradila različica 2.1. Trenutna aktualna različica

je 3.0, ki je bila objavljena leta 2010. VoiceXML je zasnovan tako, da omogoča glasovne dialoge, ki lahko vsebujejo: besedilno funkcijo TTS, predvajanje zvočnih posnetkov, prepoznavo govora oziroma glasovne vnose, vnos z DTMF-toni (tipkami telefona), snemanje glasovnih posnetkov in prevezovanje klicev. Glavni cilj VoiceXML-iniciative je uvajati spletne tehnologije v IVR-aplikacije. V VoiceXML obstajata dva tipa dialogov: forma (angl. *form*) in meni (angl. *menu*). Forme so namenjene podajanju in zbiranju informacij, meniji pa ponujajo opcije, ki omogočajo dostop do drugih dialogov. VoiceXML-dokumenti se na glasovni brskalnik prenašajo z uporabo internetnega aplikativnega protokola HTTP. Na glasovni brskalnik se prenašajo oziroma nalagajo s spletnih strežnikov. VoiceXML-dokumente nato glasovni brskalnik interpretira in izvaja. Paralelno lahko zaznava uporabnikove vnose in proži nove HTTP-zahteve na spletni strežnik. Glasovni brskalniki lahko interpretirajo več VoiceXML-dokumentov hkrati in za več hkratnih klicev. Podobno kot pri spletnih brskalnikih in spletnih strežnikih tudi pri glasovnih brskalnikih klic in HTTP-zahteve tvorijo spletno sejo, ki ima lasten podatkovni kontekst (W3C).

4.2.2 Speech Recognition Grammar

Specifikacijo za Speech Recognition Grammar je leta 2004 javno objavila organizacija W3C. Objavljen je bil skupaj z VoiceXML-različico 2.0. Format zapisa je namenjen za uporabo s prepoznavalniki govora in drugimi aplikacijami. Podpira dve obliki zapisa: XML in ABNF. XML-oblika se tipično hrani v datotekah s končnico *grxml*, uporablja pa se v vnosnih poljih VoiceXML-dokumentov, da se določi oziroma omeji možnosti vnosov. Omogoča dva načina vnosov: DTMF (toni tipk) in glasovnega (angl. *voice*), ki v vnosnem polju aktivira prepoznavo govora. V VoiceXML je mogoče uporabljati oba vnosna načina hkrati, tako da se sklicuje na dve datoteki *grxml* v enem vnosnem polju. Format je zasnovan tako, da omogoča pisanje pravil (angl. *rule*) s seznamami možnosti (angl. *one-of*) elementov (angl. *item*) prepoznave. Omogoča navzkrižno sklicevanje med pravili in dokumenti, kar omogoča programerju, da vsebino razdeli v funkcijske sklope, ki jih lahko v kontekstu ene prepoznave večkrat pouporabi. Format specificira tudi možnost uporabe značke (angl. *tag*), ki dovoljuje pisanje programske kode ali drugih navodil za prepoznavalnik govora. Vsebina značke ni natanko specificirana in jo lahko poljubno specificira proizvajalec prepoznavalnika govora (W3C).

4.2.3 HP OpenCall Media Platform

HP OCMP je glasovni brskalnik (angl. *voice browser*), ki omogoča hiter razvoj storitev tipa IVVR. Platforma je zasnovana po telekomunikacijskih standardih in jo uporablja 40 od 50

največjih svetovnih ponudnikov telekomunikacijskih storitev. Namenjena je integraciji v t. i. omrežja naslednje generacije in IP-omrežja. Za razvoj storitev HP OCMP ponuja vmesnike, ki temeljijo na odprtih standardih, kot sta VoiceXML in CCXML ter programski jezik Java. Na voljo sta tudi vmesnika MSCML in Netann, ki omogočata tretjim aplikacijam, da z uporabo protokola SIP krmilijo medijske tokove priključene na HP OCMP. Tako lahko nastopa tudi v vlogi MRF-komponente (HP 2009, 9–10).

4.2.4 Scansoft Open Speech Recognizer

Scansoft OSR je prepoznavalnik govora, ki je integriran z glasovnim brskalnikom. V našem primeru je OSR integriran s platformo HP OCMP in IVR-aplikacijo, ki uporablja VXML za glasovni uporabniški vmesnik.

Prepoznavalnik govora OSR interpretira ukaze »speech recognition grammar«, ki so lahko del VoiceXML-datoteke. V tem primeru gre za priloženo gramatiko (angl. *inline grammar*). Lahko pa se gramatični ukazi naloženi tudi iz datotek grxml, v tem primeru pa se datoteke z gramatiko nalagajo s spletnega naslova, z uporabo HTTP-protokola. Ko glasovni brskalnik interpretira VoiceXML-element, ki zahteva vnos uporabnika, posreduje ukaze gramatike na OSR. V trenutku vnosa glasovni brskalnik tudi priključi medijski tok na OSR-komponento »Speech Detector«. Ko komponenta »Speech Detector« zazna govor, sporoči glasovnemu brskalniku, da preklopi medijski tok na komponento »Speech Recognizer«, ki prične s prepoznavo govora, ki je v skladu z naloženimi gramatičnimi pravili. Rezultat prepoznave govora OSR vrne glasovnemu brskalniku, ki jih obdela v skladu z VoiceXML-dokumentom, ki ga je predhodno naložil iz spletne IVR-aplikacije (Scansoft 2005, 1–2).

4.3 Java

Oracle Java 8 je strojno neodvisna platforma, ki je sestavljena iz dveh komponent: JVM in API. JVM izvaja prevedeno javansko kodo, API pa je skupina že pripravljenih funkcij, ki so urejene v pakete in so predpripravljene in na voljo za uporabo neodvisno od strojne platforme. Za izdelavo aplikacije sem uporabil programski jezik Java, ki je med drugim visoko-nivojski, objektno orientiran, dinamičen in večniten (Oracle Docs).

4.3.1 Java Servlet

Tehnologija Java Servlet omogoča razvijalcem spletnih aplikacij enostaven in enovit način, da razširijo funkcionalnost spletnega strežnika, in dostop do obstoječih zalednih poslovnih sistemov. Mnoge spletne aplikacije uporabljajo to tehnologijo, saj omogočajo modularno zasnovu in so arhitekturno neodvisni od platforme, na kateri se izvajajo. Servleti so del

vmesnikov skupine Java EE in lahko dostopajo do drugih Java API-funkcij. Zato so servleti priljubljena izbira za izdelavo interaktivnih spletnih aplikacij. Ne izvajajo se samostojno, ampak uporabljajo komponento »servlet container«, ki je v večini primerov kombinacija spletnega in aplikacijskega strežnika, kot so to na primer: BEA WebLogic Application Server, IBM WebSphere, Sun Java System Web Server, Sun Java System Application Server in mnogi drugi (Oracle Docs).

4.3.2 Knjižnica Spring Framework

S knjižnico Spring Framework se lahko razširi privzet Java API. Izbral sem jo zato, da sem pohitрил razvoj aplikacije, saj mi mnogih funkcij, ki jih zagotavlja Spring Framework, ni bilo treba implementirati. Spring Framework vsebuje širok nabor funkcij za razvoj modernih javanskih aplikacij, med drugim omogoča vstavljanje odvisnih komponent, razvoj po vzorcu MVC in zagotavlja vmesnik JPA, ki olajša uporabo baze podatkov (Spring Framework). To so bili tudi glavni razlogi za uporabo te knjižnice.

4.3.3 Knjižnica Hibernate ORM

Knjižnica Hibernate ORM omogoča persistenco instanc objektov v relacijsko bazo podatkov. To pomeni, da so izbrane instance objektov shranjene neodvisno od stanja JVM, kar v bistvu pomeni, da so lahko prejšnje (stare) instance objektov (podatki v aplikaciji) na voljo tudi ob ponovnem zagonu aplikacije (Hibernate).

Nenazadnje pa Hibernate ORM implementira vmesnik JPA in ga je zato mogoče enostavno integrirati z JPA-vmesnikom knjižnice Spring Framework.

4.3.4 Knjižnica Thymeleaf

Knjižnica Thymeleaf je moderen strežniški stroj Java za izvajanje predlog (Thymeleaf). Omogoča različne dialekte, kot sta tudi HTML in XML. Ker VoiceXML uporablja XML-notacijo, je Thymeleaf primeren tudi za pisanje predlog za glasovni uporabniški vmesnik. Spletna aplikacija lahko tako vrača HTML- in VoiceXML-dokumente iz predlog, ki se dinamično ustvarjajo na aplikacijskem strežniku in na zahtevo iz podatkov ali modela, ki jih vrača poslovna logika ali kontroler. Podobno kot Hibernete se tudi Thymeleaf povsem enostavno integrira s knjižnico Spring Framework in v MVC-kontekstu zagotavlja funkcionalnost pogleda (angl. *view*).

4.4 Apache Tomcat

Strežnik Apache Tomcat je odprtokodna implementacija tehnologij Java Servlet, JavaServer Pages, Java Expression Language in Java WebSocket (Apache Tomcat). Lahko tudi rečemo,

da je okrnjen aplikacijski strežnik za izvajanje kode Java. Okrnjen je zato, ker ne omogoča celotnega nabora funkcij za podporo aplikacij Java EE. Pogosto se ga uporablja zato, ker omogoča izvajanje aplikacij tipa Java Servlet. Med drugimi funkcijami zagotavlja tudi podporo za HTTP-protokol in je zato tudi spletni strežnik, ki lahko streže statične in dinamične vsebine.

Apache Tomcat se lahko uporablja tudi v preverjenem okolju (angl. *validated environemnt*), ker je izdelan kot večina odprtokodnih aplikacij in programskih paketov. Ker je njegova izvorna koda javno dostopna, že v osnovi zadošča kriteriju preverljivosti. Hkrati pa je vsaka sprememba izvorne kode in binarne različice dokumentirana in javno objavljena. Iz uporabniških e-poštnih arhivov je razvidno, da podjetje Merck uporablja Apache Tomcat v preverjenem okolju. Aplikacija podjetja Interchange Digital, ki tudi uporablja Apache Tomcat, pa je bila nameščena v več farmacevtskih podatkovnih centrih (Kolinko 2011).

4.5 MySQL Community Server

MySQL Community Server je odprtokodna različica MySQL-strežnika. MySQL-strežnik je visokozmogljiv, večniten, večuporabniški sistem za upravljanje s SQL-relacijskimi bazami podatkov. Torej je MySQL strežnik DBMS in se lahko uporablja kot samostojna aplikacija ali pa integrirano v drugih aplikacijah za množično uporabo. Podatki so v MySQL-strežniku organizirani v relacijske baze podatkov, ki podatke shranjujejo v ločenih tabelah. Objekti, kot so baze, tabele, pogledi, vrstice in stolpci, predstavljajo logični model, ki je s stališča programiranja zelo fleksibilen. Pravila »ena na mnogo« in »mного na mnogo« določajo relacije med podatkovnimi polji, ki so v tabelah označeni s primarnimi in tujimi ključi. Pravila omogočajo tudi omejitve, kot so unikaten (angl. *unique*) ključ, obvezno polje, opcijsko polje in privzete vrednosti. V primeru dobro načrtovane baze podatkov lahko DBMS uveljavlja ta pravila in tako zagotavlja konsistenco podatkov, ki je neodvisna in aplikaciji skrita. SQL-jezik se uporablja za dostop in delo s podatki. Standardiziranih je več različic SQL-jezika (SQL-92, SQL:1999 in SQL:2003), ampak DBMS-sistem lahko SQL-vmesnik tudi razširi in MySQL ima nekaj specifičnih razširitev, ki olajšajo delo z DBMS (MySQL Developers).

4.6 HTML5, CSS in JavaScript

HTML5, CSS in JavaScript so spletne tehnologije, ki se izvajajo v spletnih brskalnikih in tvorijo uporabniški vmesnik spletnih aplikacij. Razlika med spletnimi stranmi, ki tudi lahko uporabljajo HTML5, in med spletnimi aplikacijami je v tem, da uporabniki spletnih aplikacij uporabljajo spletne brskalnike, da dostopajo do aplikacije, v kateri shranjujejo in urejajo

podatke, ter uporabljajo njene funkcije, ki se izvajajo na spletnem aplikacijskem strežniku. Spletni uporabniški vmesnik, ki se izvaja v spletnem brskalniku, tipično ne potrebuje osveževanja spletne strani, da opravi preverbo vnosa podatkov ali izpis podatkov. V spletnem brskalniku se prav tako izvajajo izrisi podatkov, kot so na primer tabele, grafi ali drugi grafični elementi.

HTML5 je razširitev prejšnjih HTML-različic in je vzratno kompatibilen. To sicer ne pomeni, da so vse nove funkcije tudi na voljo v brskalnikih, ki podpirajo le stare različice, ampak pomeni, da jih stari brskalniki prikažejo kot stare različice HTML. Največja razlika v uporabniških izkušnjah med različicama 4 in 5 je v uporabi značke »form«, ki uvaja nove tipe vnosov in dodatne parametre, kot so na primer novi atributi numeričnega vnosa: »step«, »min«, »max«, ki omogočajo preglednejšo uporabo vnosnega polja. Uporabniška izkušnja spletne aplikacije temelji tudi na uporabi programskega jezika JavaScript, ki uvaja dinamično interaktivno komponento uporabniške izkušnje. Je edini programski jezik, ki je privzeto vgrajen v spletne brskalnike in ga podpirajo praktično vsi spletni brskalniki. JavaScript je formalno standardiziran kot programski jezik ECMA Script, a se pogosto uporablja prvotno ime JavaScript, ki razen dela imena nima nič skupnega s programskim jezikom Java. Spletne aplikacije pa v kombinaciji s HTML5 uporabljajo še tehnologijo CSS, ki omogoča t. i. predstavitevni sloj spletne strani. CSS je odgovoren za vse, kar se nanaša na obliko in dizajn spletne strani. Stil in tema spletne strani sta tako neodvisna od vsebine. CSS določa parametre in lastnosti oblike elementov spletnih strani, prav tako tudi pravila za izris. V praksi to pomeni, da se lahko ista vsebina uporablja na različnih napravah. Oblikovna pravila lahko prerazporedijo vsebino, da doseže optimalno obliko za dano napravo (Cameron 2013, 1. pogl.).

4.6.1 Knjižnica jQuery

Knjižnica JavaScript jQuery se uporablja v mnogih spletnih aplikacijah, saj močno poenostavlja uporabo JavaScripta. To je predvsem zato, ker so spletne strani primarno dokumentno orientirane in podvržene dokumentnemu podatkovnemu modelu DOM, ki ga lahko JavaScript manipulira. Tako lahko na različne dogodke, kot so nalaganje, pritisk tipke, klik in mnoge druge, JavaScript spremeni strukturo in vsebino spletne strani brez osveževanja ali nove zahteve na spletni strežnik. JavaScript sicer sam po sebi ponuja API za manipulacijo DOM in ta API tudi uporablja knjižnica jQuery, ki je pravzaprav ovojnica (angl. *wrapper*) za privzeti JavaScript DOM API. Večina programerjev uporablja jQuery, ker menijo, da je

JavaScript programska koda na ta način preglednejša in jo je zato tudi lažje vzdrževati (Cameron 2013, 1. pogl.).

4.6.2 Knjižnica Bootstrap

Knjižnica Bootstrap je sestavljena iz komponent CSS in JavaScript, ki omogočata izdelavo odzivnih spletnih uporabniških vmesnikov. Največja prednost knjižnice Bootstrap je v tem, da omogoča enostavno izdelavo uporabniškega vmesnika, ki se lahko samodejno prilaga glede na zmogljivost naprave, zato je primeren za uporabo tako na mobilnih napravah z majhnimi zasloni kot tudi na osebnih računalnikih z velikimi zasloni. Knjižnica Bootstrap je enostavna za uporabo in zelo prilagodljiva. Za delovanje potrebuje JavaScript knjižnico jQuery. Omogoča izdelavo lastne teme, ki nadgradi in razširi privzeto obliko ter funkcionalnost (Bootstrap).

4.6.3 Knjižnica Chartist.js

Knjižnica JavaScript Chartist.js omogoča dinamičen izris grafov v spletnem brskalniku. Ponuja zelo bogat API, ki omogoča visok nivo prilagoditve. Omogoča odzivne izrise, ki se prilagajajo velikosti zaslona oziroma okna brskalnika. Oblika in stil grafa se lahko definira v CSS-datoteki, tako da je tudi v tem primeru oblika ločena od vsebine. Oblika in stil sta lahko del enotne teme spletne aplikacije. Podatke za izris se preprosto definira v metodi, ki izriše graf, in so v obliki seznama (angl. *array*). Vsaka dimenzija na seznamu podatkov predstavlja po eno funkcijo oziroma kategorijo izrisanih podatkov (Chartist.js).

5 IZBIRA BOLEZNI IN OPREDELITEV ZDRAVSTVENIH PARAMETROV

Pred pričetkom načrtovanja tehnične rešitve oziroma aplikacije sem določil, katero področje telemedicine bom uporabil za raziskavo. Po CALM-metodologiji je to faza snovanja. Na podlagi pregledane literature sem se odločil, da bom izdelal aplikacijo za spremljanje kroničnih bolezni.

Ob registraciji v aplikacijo mora uporabnik izbrati bolezen, za katero bo vnašal podatke. To je pomembno zato, da mu nato aplikacija ponudi polje za vnos relevantnih zdravstvenih parametrov. Oba uporabniška vmesnika sta zasnovana kot anketni vprašalnik, skozi katerega uporabnik vnaša vrednosti zdravstvenih parametrov oziroma podatkov.

V aplikaciji sem želel zajeti več različnih tipov odgovorov, a se kljub temu izkaže, da je vsaj za izbrane bolezni možno odgovore razporediti v le nekaj različnih tipov:

- **Zaprta vprašanja**, na katere je mogoče odgovoriti le z da in ne.
- **Lestvice**, ki subjektivno opisujejo količino, stanje oziroma počutje, veljavnost trditve in pogostost. Navedeni opisi so tudi podtipi lestvice.
- **Numerična vprašanja**, na katere je odgovor vrednost meritve z instrumentom. Na primer: odgovor na vprašanje »Kolikšen je nivo glukoze v krvi?« je vrednost 6,1, ki jo je uporabnik izmeril z napravo glukozo-meter.

Za izbor bolezni sem določil naslednje kriterije, po katerih bi izbrane bolezni privabile kar največ motiviranih uporabnikov. Kriteriji so:

1. Zdravstvena stroka priporoča redno spremljanje simptomov bolezni.
2. Bolezen mora v kratkem času raziskovanja zagotoviti čim več različnih vnosov podatkov.
3. Skupaj morajo izbrane bolezni pokriti čim več različnih tipov odgovorov.
4. Življenjski stil mora vplivati na potek bolezni.

Tako sem izbral Crohnovo bolezen (kronično vnetje črevesja), srčno popuščanje, diabetes (sladkorna bolezen) in hipertenzijo (povišan krvni tlak).

Crohnova bolezen se najpogosteje kaže kot ponavljajoča se trebušna bolečina ali krč, dlje časa trajajoča driska, ki je lahko tudi krvava, splošna utrujenost in slabo počutje. Kot simptomi se pojavljajo tudi bolečine v sklepih, razjede v ustih, hujšanje, zmanjšan tek in drugi. V razvitem svetu je približno 100 bolnikov na 100.000 prebivalcev. V Evropi je med 0,9 in 10 novih

bolnikov letno na 100.000 prebivalcev. Manjša obolevnost je Španiji, večja pa na Norveškem (Baraga 2014).

Srčno popuščanje povzročajo druge bolezni, ki okvarijo srce. Najpogosteje so to: srčni infarkt, arterijska hipertenzija in sladkorna bolezen. V Sloveniji je med 20.000 in 40.000 bolnikov s srčnim popuščanjem. Letno beležimo 5 novih bolnikov na 1.000 prebivalcev. Srčno popuščanje se kaže kot težko dihanje, sprva ob večjih telesnih naporih, otekline goleni in utrujenost. Poleg zdravil, ki zmanjšujejo obremenitev srca, se priporoča redna telesna dejavnost in zdrav življenjski slog, kar pomeni primerno telesno težo in zdravo prehrano (Vester 2014).

Poznamo dva tipa sladkorne bolezni. Za tip 2 velja, da na nastanek vplivajo dedna nagnjenja in okoljski vplivi, ki vključujejo zmanjšano telesno dejavnost in prehranjevanje s preveliko mero maščob in veliko koncentracijo ogljikovih hidratov. V Sloveniji ima 6,9 % ljudi sladkorno bolezen tipa 2. Število bolnikov pa se v svetu vsakih 15 let podvoji. Bolnik mora skrbeti za samokontrolo glukoze v krvi, da lahko njegov zdravnik uravnava količino zdravil. Poleg tega naj bi bolnik na vsaki stopnji bolezni imel možnost samokontrole glukoze v krvi, saj bo na ta način bolje razumel svojo bolezen ter vpliv hrane, telesne dejavnosti in zdravil na krvni sladkor (Strouhal 2014).

Pri zvišanem krvnem tlaku v 90 % primerov pravi vzrok za nastanek bolezni ni znan. Znano je, da na razvoj vplivajo dednost in dejavniki okolja, prekomerna telesna teža, pomanjkanje gibanja in sol. Bolezen je zelo pogosta. V Evropi jo ima 44 % ljudi. V določenih primerih je že pri postavitvi diagnoze potrebno domače merjenje krvnega tlaka in prav tako v primeru diagnoze. Spremljanje krvnega tlaka v domačem merjenju je ključno za razumevanje in obvladovanje bolezni (Petek-Šter 2014).

Za spremljanje navedenih bolezni bodo uporabniki lahko v aplikacijo vnašali podatke za 19 izbranih parametrov (glej Tabelo 5.1).

Tabela 5.1: Parametri za spremljanje bolezni

#	IME	TIP
1	Telesna teža	Numerični
2	Nivo glukoze v krvi	Numerični
3	Počutje	Lestvica (počutje)
4	Utrujenost	Lestvica (količina)
5	Nezadovoljstvo	Lestvica (pogostost)
6	Občutek potrebe	Lestvica (pogostost)
7	Sproščенost	Lestvica (pogostost)
8	Težave z vetrovi	Lestvica (pogostost)
9	Utrujenost	Zaprto vprašanje
10	Občutek depresije	Zaprto vprašanje
11	Bolečine v trebuhu	Zaprto vprašanje
12	Kri v blatu	Zaprto vprašanje
13	Težko dihanje	Lestvica (pogostost)
14	Tiščanje na vodo	Lestvica (pogostost)
15	Kašelj ob naporu	Lestvica (količina)
16	Hladne okončine	Zaprto vprašanje
17	Zgornji krvni tlak	Numerični
18	Spodnji krvni tlak	Numerični
19	Gibanje	Lestvica (količina)

6 NAČRTOVANJE, IZDELAVA IN PREDSTAVITEV APLIKACIJE

6.1 Načrtovanje aplikacije

Aplikacija mora omogočati vnos, urejanje in izpis podatkov za spremljanje izbranih kroničnih bolezni. V osnovi so vhodni podatki različnih tipov. Lahko so cela števila, decimalne vrednosti in besede. Ker lahko besede predstavljajo vrednosti na lestvicah odgovorov, lahko aplikacija vse podatke shranjuje kot decimalne vrednosti. Aplikacija mora imeti dva uporabniška vmesnika: spletnega in glasovnega. Glasovni uporabniški vmesnik mora omogočati le vnos podatkov za spremljanje bolezni z uporabo glasu oziroma govora in tipk telefona oziroma DTMF-tonov.

Spletni uporabniški vmesnik pa mora omogočati naslednje funkcije:

- registracijo novih uporabnikov (ustvarjanje novega uporabniškega profila),
- vnos podatkov za spremljanje kroničnih bolezni v obliki čarovnika, ki zahteva le vnos manjkajočih podatkov,
- urejanje podatkov za spremljanje kroničnih bolezni,
- izpis podatkov za spremljanje kroničnih bolezni,
- urejanje uporabniškega profila,
- urejanje uporabnikov oziroma administracija uporabnikov.

Aplikacija mora uporabniku omogočati izbiro ene ali več bolezni. Pri vnosu pa mora zagotoviti, da se enaka vprašanja ali vnosna polja za vnos podatkov ne ponavljajo. Na primer: če uporabnik izbere bolezni diabetes in hipertenzija, ki obe zahtevata podatek o telesni teži, ponudi takrat aplikacija vnosno polje za vnos podatka o telesni teži le enkrat.

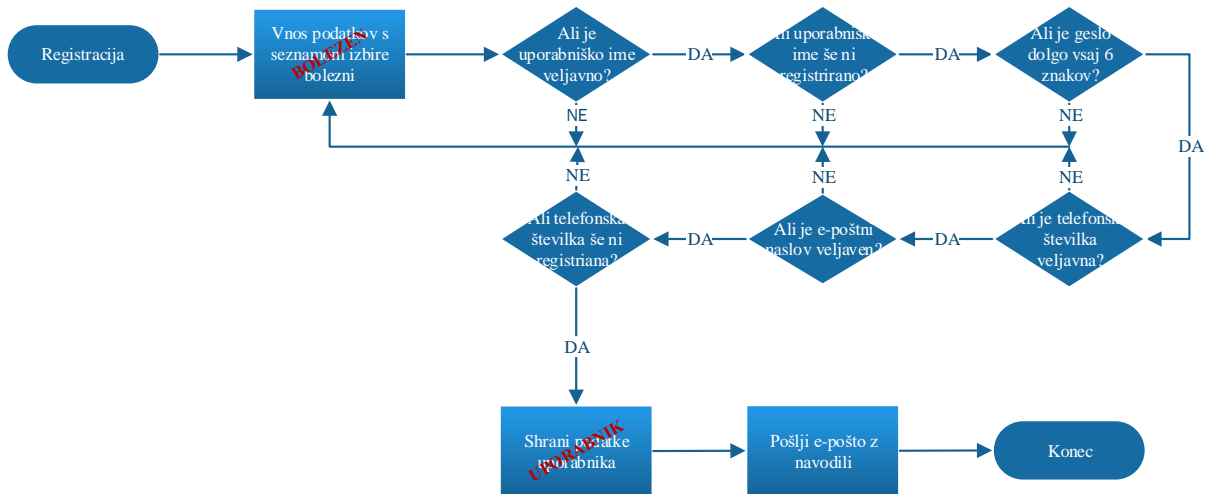
Aplikacija podatke shranjuje v relacijsko bazo podatkov, ki služi kot podatkovno skladišče, iz katerega aplikacija pridobiva podatke za izpis in ugotavlja, katere podatke je treba v danem trenutku vnesti. Za realizacijo zahtev sem identificiral štiri ključne procese:

- registracijo,
- prijavo,
- vnos podatkov,
- izpis podatkov.

V procesu registracije mora uporabnik vnesti uporabniške podatke, kot so uporabniško ime, geslo, ime, priimek, telefonska številka, e-poštni naslov, datum rojstva in spol. Izbrati pa mora

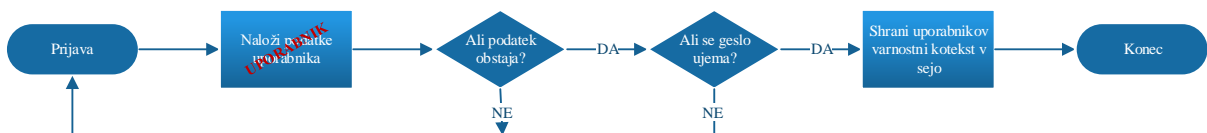
tudi bolezn. Aplikacija mora preveriti veljavnost vnosov in te podatke shraniti v bazo podatkov.

Slika 6.1: Proces registracije



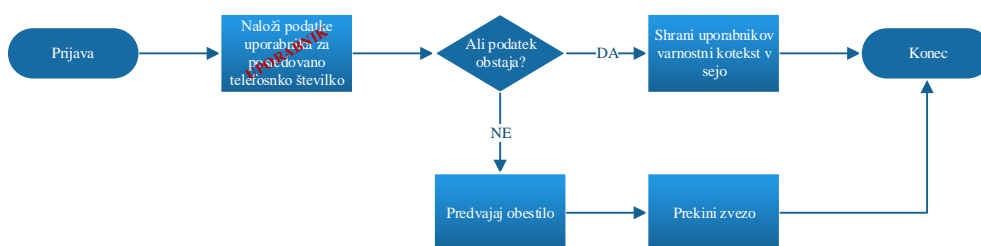
Zaradi razlik v zmožnostih uporabniških vmesnikov in udobne uporabniške izkušnje sem moral proces prijave razdeliti na dva procesa: spletno prijavo in telefonsko prijavo. V procesu spletne prijave mora uporabnik vnesti uporabniško ime in geslo. Aplikacija iz baze podatkov naloži uporabniški profil in če profil obstaja, preveri, ali se geslo ujema s shranjenim geslom (glej Sliko 6.2).

Slika 6.2: Proces prijave – splet



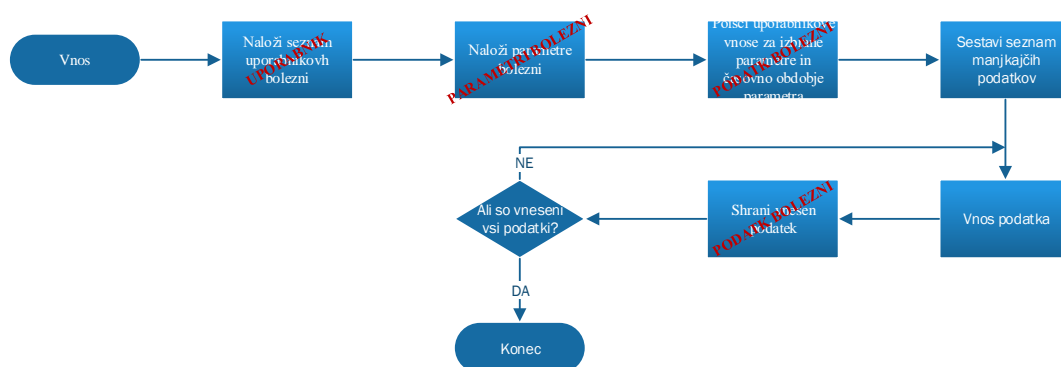
V procesu telefonske prijave pa aplikacija na podlagi telefonske številke naloži uporabniški profil. V primeru, da uporabnik obstaja, ga prijavi, v nasprotnem primeru pa predvaja obvestilo, da je treba telefonsko številko registrirati na spletni strani, in prekine zvezo (glej Sliko 6.3).

Slika 6.3: Proces prijave – telefon



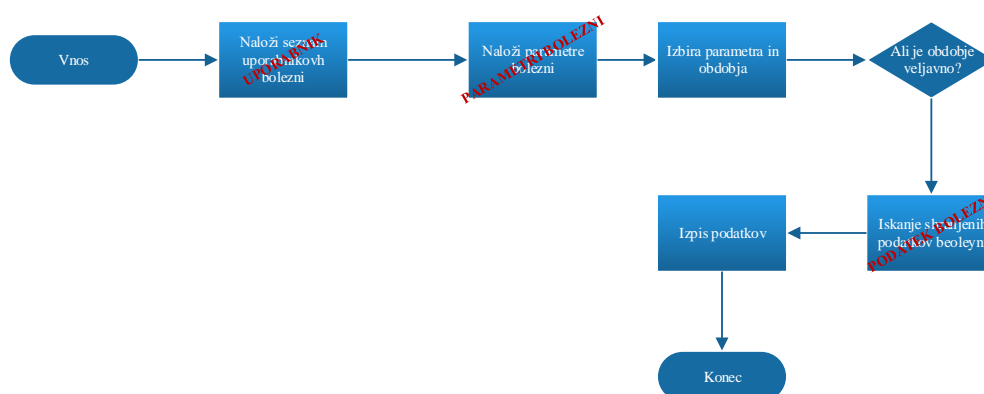
V procesu vnosa podatkov mora aplikacija preveriti, katere bolezni ima uporabnik izbrane, kateri parametri so potrebni za spremljanje uporabnikovih bolezni in katere podatke je treba v danem trenutku vnesti. Aplikacija tako sestavi seznam parametrov bolezni, ki jih mora uporabnik vnesti. Proces ni končan, dokler uporabnik skozi iteracije vnosov ne vnese vseh podatkov. Lahko pa se zgodi, da uporabnik prekine z delom, zato se po vsakem vnosu vnoseni podatek shrani. Če uporabnik prekine z delom, se odjavi ali prekine klic, mu tako ob ponovni prijavi in vnosu ni treba vnesti že vnesenih in shranjenih podatkov.

Slika 6.4: Proces vnosa podatkov



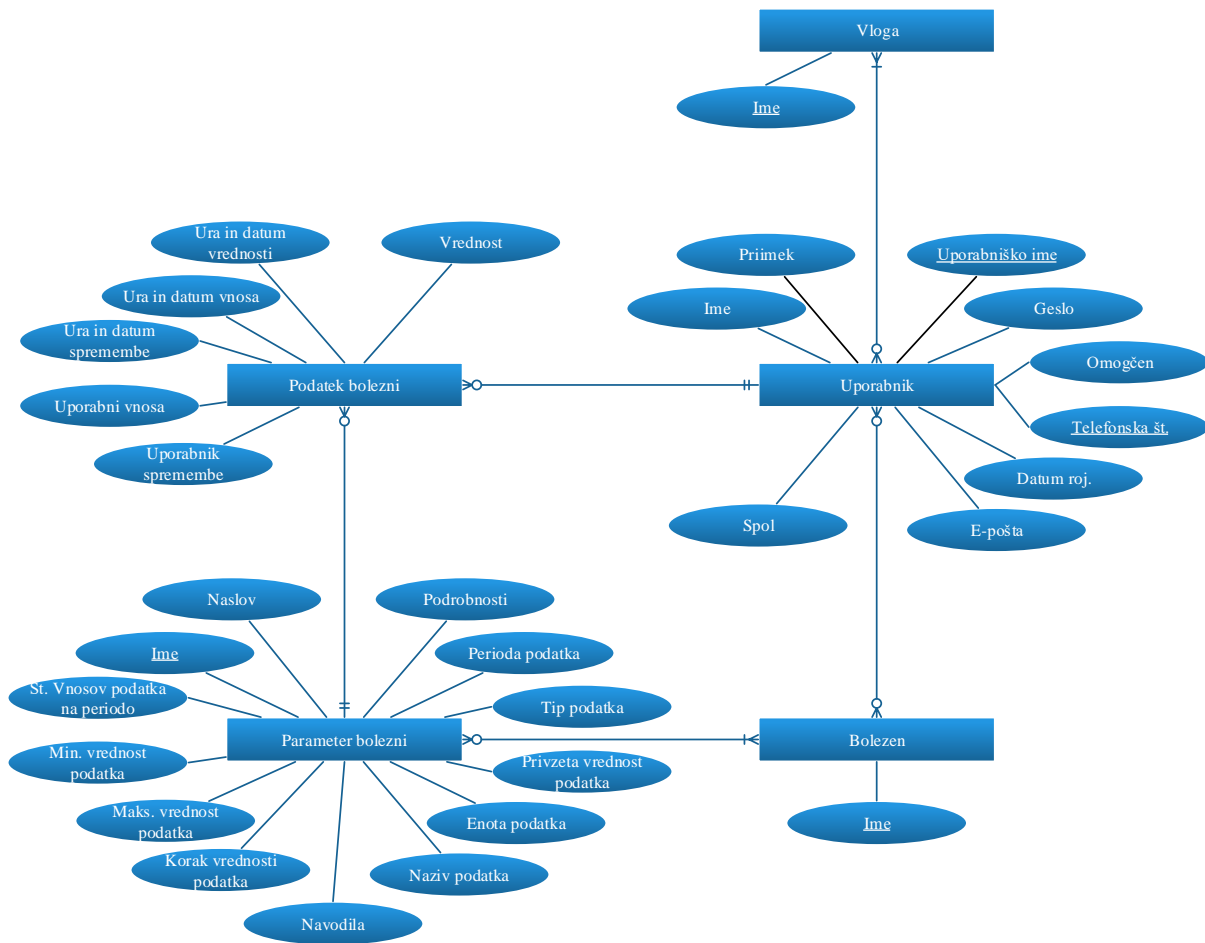
Proces izpisa podatkov je namenjen kronološkemu pregledu vrednosti, zato zahteva izbiro parametra bolezni in časovni okvir.

Slika 6.5: Proces izpisa podatkov



Z definiranjem ključnih procesov sem identificiral entitetne tipe in attribute ter izdelal diagram entiteta-relacija (ER) (glej Sliko 6.6). Poleg identificiranih entitetnih tipov: uporabnik, bolezen, parameter bolezni in podatek bolezni, sem dodal še entitetni tip vloga. Z entitetnim tipom vloga bom lahko razlikoval med vlogami uporabnikov. Uporabnik ima lahko vlogo uporabnika ali administratorja ali obe (uporabnik in administrator).

Slika 6.6: ER-diagram baze podatkov



Iz tega sledi, da ima uporabnik nič ali mnogo bolezni in prav tako ima bolezen nič ali mnogo uporabnikov. Predvideti je treba situacijo, v kateri imamo uporabnika – administratorja, ki ni nima izbrane bolezni, in situacijo, ko bolezen še nima nobenega uporabnika.

Uporabnik ima vsaj eno ali mnogo vlog. Vloga pa ima nič ali mnogo uporabnikov. To pomeni, da podatkovni model dovoljuje situacijo, ko še ni registriranih uporabnikov. Hkrati pri registraciji zahteva vnos vsaj ene vloge. Poslovna logika spletne registracije mora tako samodejno nastaviti vsaj eno vlogo ob vpisu entitete uporabnika v bazo podatkov.

Bolezen ima nič ali mnogo parametrov bolezni. Parameter bolezni pa ima eno ali mnogo bolezni. Tako lahko aplikacija iz podatkovnega modela enostavno sestavi seznam parametrov bolezni, potrebnih za vnos. Hkrati pa baza podatkov skrbi zato, da ne ostanejo shranjene entitete bolezni, kadar odstranimo še zadnjo bolezen, ki ji pripada parameter bolezni.

Parameter bolezni ima nič ali mnogo podatkov bolezni. Podatki bolezni hranijo vrednosti in druge metapodatke o vnosu podatka bolezni. Podatkovni model dopušča možnost, ko za parameter bolezni še ni vnesenega podatka bolezni. Podatek pa mora imeti natanko en

parameter bolezni. Hkrati pa ima podatek bolezni natanko enega uporabnika. Uporabnik pa ima nič ali mnogo podatkov bolezni. V bistvu je podatek bolezni le vrednost, ki jo vnese uporabnik, in skupaj s parametrom bolezni in uporabnikom tvori ključno informacijo.

6.2 Izdelava aplikacije

Izdelal sem aplikacijo telezdravje.si, ki ustreza zahtevam (glej poglavje 5.1). Aplikacijo sem izdelal z razvojnim orodjem Oracle Netbeans 8. Orodje je brezplačno in omogoča razvoj aplikacij za operacijske sisteme Windows, Linux, Mac in druge. Omogoča razvoj aplikacij v programskih jezikih Java, PHP, C in C++ (Oracle).

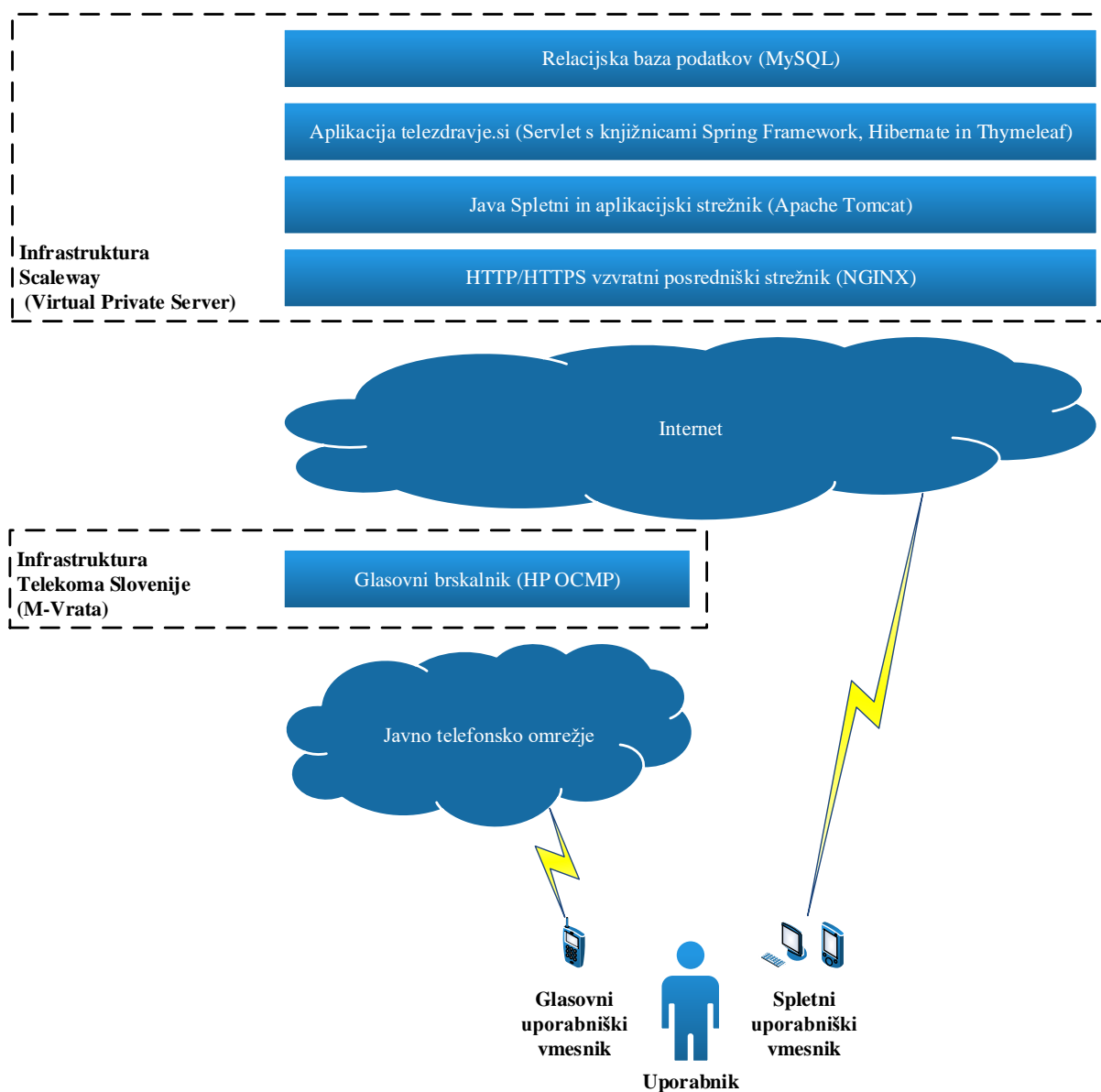
6.2.1 Konceptualna zasnova

Uporabniki dostopajo do aplikacije telezdravje.si iz dveh virov:

- **S telefonskimi aparati** prek javnega telefonskega omrežja do glasovnega brskalnika, ki je del infrastrukture Telekoma Slovenije (M:Vrata). To je platforma HP OpenCall Media Platform, ki prek omrežja internet nalaga dokumente VXML in Speech Recognition Grammar ter zvočne posnetke s spletno-aplikacijskega strežnika, ki deluje pri ponudniku Scaleway.
- **S spletnimi brskalniki** prek omrežja internet dostopajo do spletnega naslova <http://telezdravje.si>, ki je dostopen na spletno-aplikacijskem strežniku, ki deluje pri ponudniku Scaleway. Spletni brskalniki nalagajo, interpretirajo in izvajajo dokumente in knjižnice HTML, CSS in JavaScript.

V obeh primerih se na navideznem privatnem strežniku (angl. *Virtual Private Server*) v popolnoma enakih komponentah izvajajo HTTP-zahteve uporabnikov (glej Sliko 6.7). Zahteve najprej vstopijo v vzratni HTTP-posredniški strežnik (angl. *reverse proxy*) Nginx, ki jih posreduje naprej na spletno-aplikacijski strežnik Apache Tomcat. Ta na podlagi zahtevanega internetnega naslova poišče aplikacijo Java Servlet in ji preda zahtevo v izvajanje. Aplikacija telezdravje.si, ki vključuje knjižnico Spring Framework, ovrednoti zahtevano spletno pot in izvede poslovno logiko zahtevane spletne poti, ki je del spletnega naslova. Rezultat izvajanja je pogled, ki na podlagi vhodnih podatkov in predloge vrne ustrezno oblikovan dokument HTML5 ali VoiceXML (odvisno od zahteve). Izvajanje pogledov je del knjižnice Thymeleaf. V primeru, da mora poslovna logika dostopati do starih podatkov, jih spremeniti ali shraniti nove, pa uporabi knjižnico Hibernate ORM. V aplikaciji se SQL-stavki neposredno ne uporabljajo. SQL-stavke in poizvedbe tvori knjižnica Hibernate. Aplikacija pa uporablja višjenivojski vmesnik JPA, ki za poizvedbe uporablja jezik HQL.

Slika.6.7: Konceptualna zasnova rešitve telezdravje.si



6.2.2 Podatkovni model

V spodnji tabeli so prevodi uporabljenih imen tabel oziroma entitetnih tipov. Za izdelavo aplikacije (programiranje) sem namreč uporabljal angleški jezik (glej Tabelo 6.1).

Tabela 6.1: Preslikava entitetnih tipov v tabele baze podatkov

ENTITETNI TIP	TABELA
Uporabnik	User
Vloga	Role
Bolezen	Medical condition
Parameter bolezni	Medical parameter

Podatek bolezni	Medical data
-----------------	--------------

ER-diagram (glej Sliko 6.6), ki predstavlja relacije med entitetnimi tipi s pripadajočimi atributi, je treba bolj konkretno opredeliti, da ga je mogoče uporabiti v DBMS. Med vse relacije entitetnih tipov, ki zahtevajo implementacijo relacije mnogo na mnogo, je treba dodati vezne tabele. Za izvajanje varnostne politike, ki temelji na pravicah, ki so dodeljene v vloge, sem moral dodati še entitetni tip Pravica. Dodal sem tudi umetne primarne ključne (id), da sem lahko v programski kodi ustvaril generični entitetni tip oziroma razred Java s ključnimi atributi in metodami, ki so ga nato ostali bolj specifični razredi lahko dedovali. Spodnja fizična shema predstavlja celoten konkretni tehnični podatkovni model baze podatkov, ki jo uporablja aplikacija telezdravje.si (glej Sliko 6.8).

podatke, ki jih uporablja v spletnih pogledih, ki se izrisujejo v spletnih brskalnikih. Pogledi se nahajajo v mapi /src/main/webapp/WEB-INF/templates/html in poleg kode HTML5 vsebujejo tudi navodila za Thymeleaf. Ko Thymeleaf izvaja pogled, ta pravila, ki so v večini v atributih HTML-elementov z imenskim prostorom »th«, po obdelavi pretvori v veljavne HTML-značke in jih odstrani. Ti dodatni atributi omogočajo dinamično generiranje dokumentov HTML5 iz podatkovnega modela, ki ga nastavlja kontroler. Ti podatki so lahko na primer vprašanja za vnos zdravstvenih parametrov ali pa sezname podatkov JavaScript za izris grafikona.

Vse strani HTML5, ki jih vrača aplikacija telezdravje.si, vključujejo knjižnici Bootstrap in jQuery. Predloge Thymeleaf, ki zahtevajo vnos podatkov, pa vsebujejo tudi kodo JavaScript, ki omogoča preverjanje podatkov, preden se posredujejo na aplikacijski strežnik v spletno aplikacijo.

Za spletni uporabniški vmesnik sem izdelal visoko kontrastno temo CSS, ki spremeni privzeti izgled teme Bootstrap. Izvorna koda za prilagoditve je shranjena v datotekah less v mapi /src/main/resources/less. Datoteke less se v procesu gradnje prevedejo v CSS, ki se po zaključeni gradnji namesti v datoteko theme.css in postane sestavni del statičnih datotek in knjižnic. Pogledi Thymeleaf se tako sklicujejo na te datoteke in knjižnice in so zato vsebovane v kodi HTML5, ki jih aplikacija ob zahtevah HTTP vrača spletnim brskalnikom.

6.3.1 Programiranje glasovnega uporabniškega vmesnika

Programiranje glasovnega vmesnika je videti zelo podobno kot programiranje spletnega vmesnika. Vsa poslovna logika glasovnega vmesnika se nahaja v paketu si.bvukic.telehealth.ivr.controller. Glasovni brskalnik pošilja zahteve na HTTP-naslove, ki vsebujejo poti do metod razredov v omenjenem paketu. Na HTTP-zahteve aplikacija vrača dokumente VoiceXML, ki so prav tako kot HTML5 generirani iz predlog Thymeleaf. Princip delovanja je popolnoma enak kot pri spletnem uporabniškem vmesniku. Predloge za vsebine VoiceXML se nahajajo v mapi /src/main/webapp/WEB-INF/templates/xml/vxml.

Eden večjih izzivov te naloge je bilo implementiranje prepoznavalnika govora. VoiceXML ponuja le integracijski vmesnik za gramatiko prepoznavalnika govora (angl. speech recognition grammar), v katerem je treba definirati vsa pravila za prepoznavo govora. Z vidika VoiceXML je aplikacija zelo enostavna, saj uporablja le dialoge tipa forma. To je mogoče zato, ker je glasovni uporabniški vmesnik zelo enostaven in uporabniku ne omogoča menijske izbire. Uporabniški vmesnik je zasnovan kot čarovnik, ki uporabnika vodi skozi vprašanja oziroma vnose (glej prilogo B).

Za prepoznavo govora se v aplikaciji uporabljajo datoteke grxml. Grxml za prepoznavo vprašanj tipa lestvica in zaprta vprašanja je zelo enostaven. Enako velja za DTMF-prepoznavo. Datoteka z gramatiko za prepoznavnik govora za numerični tip vprašanj pa vsebuje več pravil, navzkrižno sklicevanje in programsko logiko, napisano v programskem jeziku ECMA Script. Tako lahko ena statična datoteka grxml omogoča prepoznavo števil od 0 do 9.999,9999 (glej prilogo B).

6.4 Predstavitev aplikacije

Aplikacijski paket je nameščen na strežnik pri ponudniku Scaleway. Uporabnikom je aplikacija dostopna na internetnem naslovu <http://telezdravje.si> in na telefonski številki 051 951 155. Zaradi odzivne CSS-teme lahko uporabniki do spletnega uporabniškega vmesnika dostopajo tudi z mobilnih naprav in ne le z osebnega računalnika. Sistemske zahteve za uporabo aplikacije prek spletnega uporabniškega vmesnika so:

- dostop do interneta,
- eden izmed naštetih brskalnikov: Microsoft Internet Explorer 10 (ali novejši), Google Chrome 40 (ali novejši), Mozilla Firefox 43 (ali novejši), Apple Safari 8 (ali novejši) in drugi sodobni brskalniki, ki temeljijo na tehnologiji WebKit.

Za uporabo glasovnega (IVR) vmesnika pa je zahtevan le telefonski aparat, priključen v javno telefonsko omrežje.

6.4.1 Spletni uporabniški vmesnik

Ko se uporabnik poveže s spletnim naslovom aplikacije, se mu odpre prijavno okno, v katerega lahko vpiše uporabniško ime in geslo, in se tako prijavi v spletni uporabniški vmesnik. Prek prijavnega okna pa lahko dostopa tudi do obrazca za registracijo in do kratkih navodil, kjer je zapisana tudi telefonska številka glasovnega uporabniškega vmesnika.

Registracijski obrazec je izdelan tako, da preverja vnesene podatke in uporabniku sporoča morebitne napake. Za uspešno registracijo mora uporabnik izbrati vsaj eno bolezen (glej Sliko 6.9).

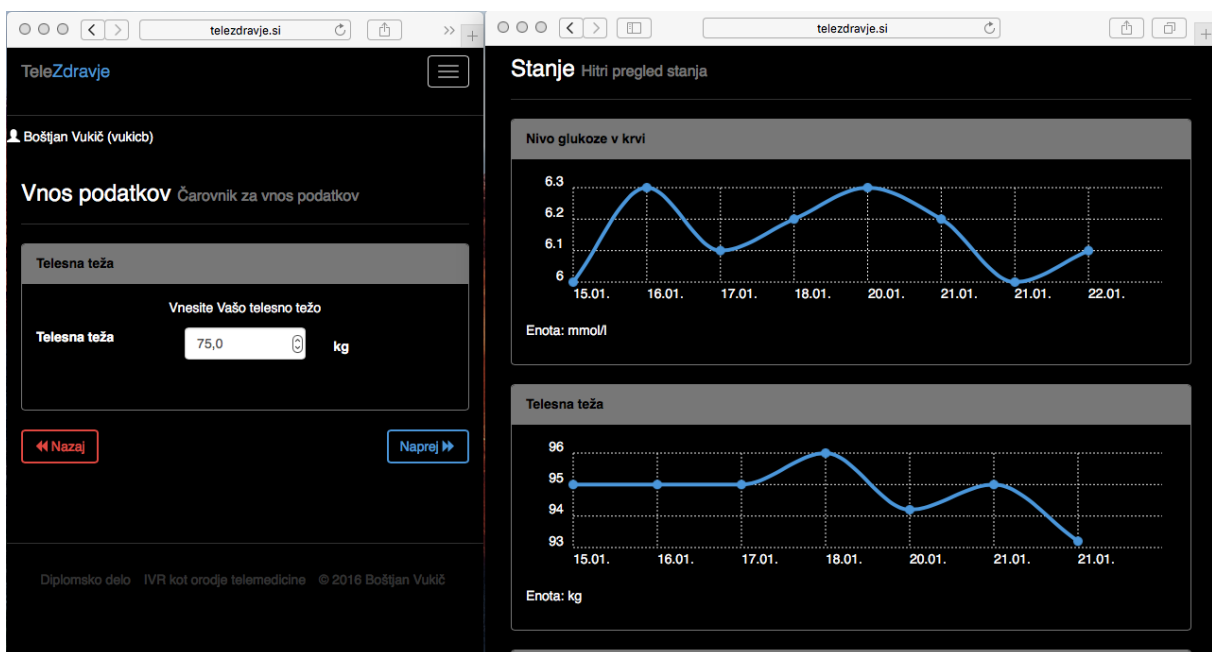
Slika 6.9: Prijava in registracija

The image shows two browser windows from the website telezdravje.si. The left window displays the login page with fields for 'Uporabniško ime' and 'Geslo', a 'Prijavi se' button, and a link for 'Potrebujete pomoč?'. The right window displays the registration page with a 'Registracija' header and a 'Vpišite uporabniške podatke' instruction. It contains a 'Podatki' form with fields for 'Uporabniško ime' (boštjanvukic), 'Geslo', 'Potrdi geslo', 'Ime' (Boštjan), 'Priimek' (Vukič), 'E-Pošta' (boštjan@example.tld), 'Telefon' (011231234), 'Datum rojstva' (06.06.1982), and 'Spol' (moški). A 'Bolezni' section has checkboxes for 'Crohnova bolezen', 'Diabetes', 'Srčno popuščanje', and 'Hipertenzija'. A 'Registriraj' button is at the bottom.

Po uspešni registraciji se uporabniku izpišejo navodila za uporabo, ki jih prejme tudi na registrirani e-poštni naslov. Uporabnik lahko takoj nadaljuje s prijavo.

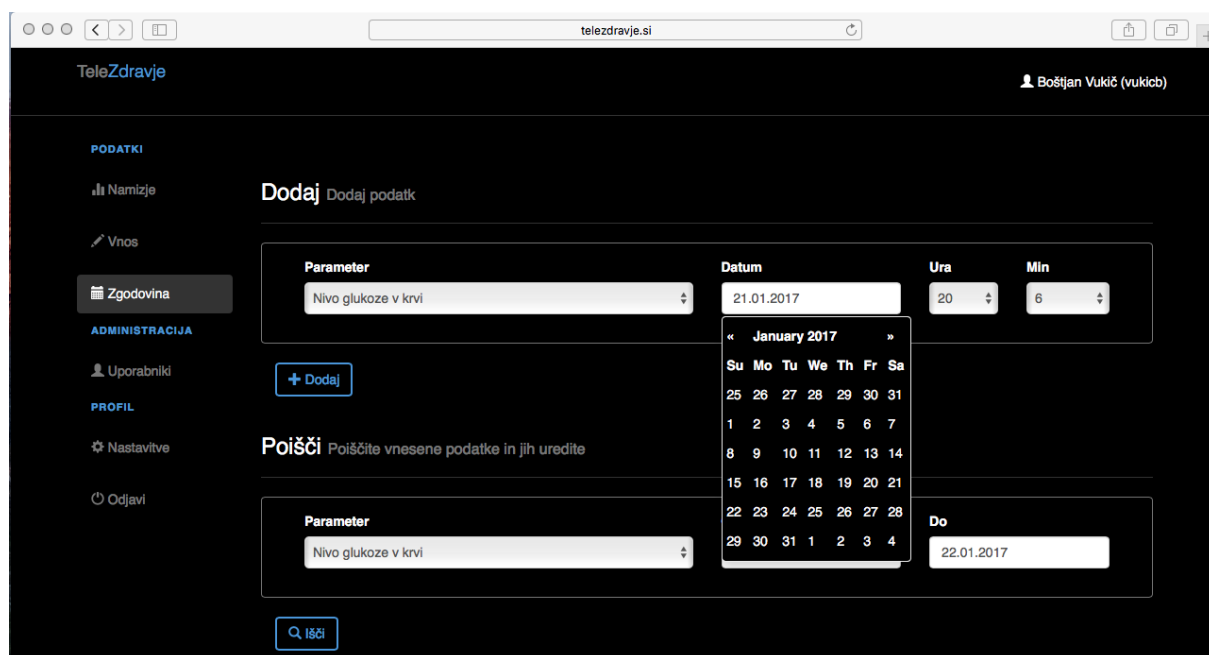
Po uspešni prijavi se uporabniku naloži namizje, ki v grafih izpiše vnesene podatke. Za nov vnos lahko uporabnik uporabi meni Vnos. Vnos je izdelan tako, da uporabi poslovno logiko, ki preveri, katere podatke je treba vnesti. Uporabnik je tako voden skozi vsa vnosna polja, za vse bolezni, ki jih je navedel ob registraciji in jih je v danem trenutku treba vnesti.

Slika 6.10: Vnos in namizje



Vnos pa ne omogoča dodajanja, spreminjanja in brisanja podatkov za obdobja iz zgodovine. Za urejanje zgodovine je uporabniku na voljo meni Zgodovina. Za dodajanje mora uporabnik izbrati parameter, datum in čas. Če želi uporabnik urejati podatke, jih mora najprej poiskati. Za to mu je na voljo iskalnik, v katerem izbere parameter, ki ga želi urejati, ter začetni in končni datum za omejitev rezultatov iskanja. Če so v izbranem obdobju za izbrani parameter shranjeni podatki, jih aplikacija izpiše v tabeli, ki vsebuje vrednosti: datum in čas, ime parametra, vrednost ter gumba za urejanje in brisanje podatka.

Slika 6.11: Zgodovina



6.4.2 Glasovni uporabniški vmesnik

Glasovni uporabniški vmesnik je zelo preprost in uporablja enako poslovno logiko za iskanje manjkajočih podatkov, kot jo uporablja spletni uporabniški vmesnik. Ko uporabnik pokliče številko IVR-sistema, se v zahtevi v aplikacijo posreduje tudi uporabnikova telefonska številka. Sistem preveri, ali je številka registrirana, in če je registrirana, potem uporabniku predvaja pozdravno sporočilo: »Dobrodošli v sistemu telezdravje!«. V nasprotnem primeru – če telefonska številka ne pripada registriranemu uporabniku –, pa sistem predvaja ustrezno sporočilo z navodili za registracijo prek spletnega uporabniškega vmesnika.

Če telefonska številka pripada registriranemu uporabniku, sledi obvestilo: »Sistem bo preveril, katere podatke je treba vnesti, in vas pozval, da jih vnesete.«. V tem trenutku poslovna logika preveri v bazi podatkov, kateri podatki manjkajo, in prične z izpraševanjem.

Ko uporabnik odgovori na vsa vprašanja, se mu predvaja sporočilo: »Vsi vaši podatki so posodobljeni. Nasvidenje.« in zveza se prekine.

Prek glasovnega vmesnika lahko uporabnik vnese vse podatke, zato mu aplikacija v fazi izpraševanja (vnosa) lahko zastavi vprašanja, ki so tipizirana in podprta za vnos s prepoznavo govora in tipk telefona.

Če prepoznavnik govora ne zazna ali ne more določiti vnesene vrednosti, se uporabniku predvajajo ustrezne napake in navodila, ki pojasnijo možnosti glasovnega vnosa in vnosa s tipkami telefona. Navodila so tako vezana na vnosne možnosti, ki so posledica tipa vprašanja oziroma tipa zdravstvenega parametra.

Tabela 6.2: Vprašanja v glasovnem vmesniku

VPRAŠANJE / POZIV ZA VNOS
Izgovorite telesno težo.
Izgovorite nivo glukoze v krvi.
Kako se počutite?
Kako pogosto ste utrujeni?
Kako pogosto nezadovoljni?
Imate pogosto občutek potrebe?
Ste pogosto sproščeni?
Imate težave z vetrovi?
Imate pogosto občutek depresije?
Imate bolečine v trebuhu?
Imate kri v blatu?
Kako pogosto težko dihate?
Vas ponoči pogosto tišči na vodo?
Koliko kašljate ob naporu?
Imate hladne okončine?
Koliko vam otekajo noge?
Izgovorite spodnji krvni tlak.
Izgovorite zgornji krvni tlak.
Koliko se gibate?

7 SKLEP

V diplomskem delu sem na študiji primera pokazal, kako je mogoče združiti spletni in glasovni uporabniški vmesnik v eno spletno aplikacijo. Pristop, ki vključuje tudi glasovni uporabniški vmesnik, ima potencial, da doseže tudi populacijo, ki ni digitalno pismena in ne uporablja interneta. Aplikacija telezdravje.si dokazuje tudi, da je mogoče zasnovati in izdelati glasovni uporabniški vmesnik za potrebe spremljanja zdravja kroničnih bolnikov. Glasovni uporabniški vmesnik uporablja izključno odprte in standardizirane tehnologije, kot sta VoiceXML in Speech Recognition Grammar, ki se v zadnjih dvanajstih letih niso bistveno spremenile. Leta 2010 je bil objavljen nov standard VoiceXML, različica 3.0, ki poleg podpore za videovsebine prinaša precej novosti s področja izvajanja vsebine. V osnovi pa format zapisa in glasovne funkcije ostajajo primerljive z različicama 2.0 in 2.1. To dokazuje, da je tehnologija zrela. Ponudba glasovnih brskalnikov, kot je HP OCMP, je velika. Podobne produkte izdelujejo tudi druga večja podjetja, kot so Avaya, Cisco, IBM in Voxeo. IVR-rešitve, ki temeljijo na omenjenih odprtih standardih, so zato lahko cenovno ugodne in so lahko dolgoročna naložba. Tako so IVR-aplikacije idealne za uporabo v javnem sektorju in večjih sistemih, ki se počasi odzivajo na spremembe.

Z vidika uporabnosti je glasovni uporabniški vmesnik primeren za vnose, ki imajo omejen nabor možnosti. Na ta način se izboljša natančnost prepoznave. V vsakem primeru pa mora glasovni uporabniški vmesnik uporabniku omogočati, da preveri in potrdi vnos. Glasovni uporabniški vmesnik je namreč prikrajšan za vizualno komponento, ki uporabnikom spletnega uporabniškega vmesnika omogoča preverjanje že sproti, ob vnašanju podatkov. Natančnost vnosov sem pri uporabnikih preveril na podlagi telefonskega pogovora. Večina uporabnikov je opazila slabšo natančnost pri glasovnem vnašanju številskih vrednosti od 10 do 99. Ob tem pa z DTMF-vnosi niso imeli težav. Iz uporabniške izkušnje teh nekaj uporabnikov ni mogoče sklepati na širšo populacijo, kot je na primer populacija državljanov Slovenije. Je pa ta študija primera pokazatelj, da je ustrezno načrtovan glasovni vmesnik primeren za vnos celih in tudi decimalnih števil in ne le posamičnih števil. Uporabniško izkušnjo pri prepoznavi govora bi se dalo izboljšati z uporabo datoteke leksikon (angl. *lexicon*), saj sistem OSR omogoča definiranje lastnih besed oziroma ključev, ki vsebujejo fonetični zapis možnega vnosa. Na ta način je mogoče podpreti različne dialekte, domače in pogovorne izgovorjave besed. Izdelava leksikona in študija uporabnosti na statistično reprezentativnem vzorcu bi lahko bila cilja nadaljnjega raziskovanja, ki bi znanstveno dokazalo primernost in pomembnost uporabe IVR-aplikacij pri osebah, starejših od 45 let.

Glede na aktivnosti predstavljenega slovenskega projekta e-zdravje bi bilo smotrno kot sestavni del sistema za eNaročanje in eReceipt vključiti tudi IVR. Tako bi lahko naredili funkcije omenjenih sistemov in storitev dostopnejše dobri petini Slovencev. Obenem bi lahko uvedli tudi dodatne funkcije, ki so na voljo le prek glasovnega odzivnika, kot so na primer napotki in količine jemanja zdravil, samodejno telefonsko obveščanje o spremembah naročenih terminov in opomniki ter morebitni napotki o naročenem zdravstvenem posegu ali pregledu. V tem kontekstu bi IVR resnično postal orodje telemedicene, saj bi bil integrirani del medicinskega informacijskega sistema.

8 LITERATURA

1. Aleksic, Petar, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach in Pedro Moreno. 2015. *Bringing Contextual Information to Google Speech Recognition*. Dostopno prek: <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43819.pdf> (13. december 2016).
2. *Apache Tomcat*. Dostopno prek: <http://tomcat.apache.org> (23. januar 2017).
3. Baraga, Dušan. 2014. *Crohnova belezen: Navodila za bolnike*. Dostopno prek: <http://www.drmed.org/wp-content/uploads/2014/06/II-141.pdf> (29. april 2016).
4. Bashshur, Rashid L. in Gary W. Shannon. 2009. National Telemedicine Initiatives: Essential to Healthcare Reform. *Telemedicine and e-Health* 15 (6): 600–610.
5. *Bootstrap*. Dostopno prek: <http://getbootstrap.com/> (19. april 2016).
6. Cameron, Dane. 2013. *A Software Engineer Learns HTML5, JavaScript and jQuery: A guide to standards-based web applications*. Wellington: Cisdal Publishing. Dostopno prek: Amazon Kindle.
7. Charness, Neil, George Demiris in Elizabeth A. Krupinski. 2012. *Designing Telehealth for an Aging Population: A Human Factors Perspective*. Boca Raton: CRC Press.
8. *Chartist.js*. Dostopno prek: <https://gionkunz.github.io/chartist-js> (7. december 2016).
9. *Cloud Speech API*. Dostopno prek: <https://cloud.google.com/speech/> (27. december 2016).
10. Eadicicco, Lisa. 2017. *Amazon Is Already Winning the Next Big Arms Race in Tech*. Dostopno prek: <http://time.com/4624067/amazon-echo-alexa-ces-2017/> (13. januar 2017).
11. Fong, Bernard, A.C.M. Fong in C.K. Li. 2010. *Telemedicine Technologies: Information Technologies in Medicine and Telehealth*. Chichester: John Wiley & Sons Ltd.
12. Gostiša, Nataša. 2016. *Elektronske komunikacijske storitve, Slovenija, 4. četrletje 2015*. Dostopno prek: <http://www.stat.si/StatWeb/prikazi-novico?id=5820&idp=25&headerbar=16> (21. marec 2016).
13. *Hibernate*. Dostopno prek: <http://hibernate.org> (21. januar 2017).
14. HP. 2009. *HP OpenCall Media Platform 4.2 Product Description*. Palo Alto: HP.

15. Hura, Susan. 2012. VUI Designers Know Cross-Channel Design. *Speech Technology* (november/december): 4.
16. IBM. 2008. *Collaborative Application Lifecycle Management with IBM Rational Products*. North Castle: IBM Corp.
17. Kolinko, Konstantin. 2011. *FDA (21 CFR Part 11) Validation*. Dostopno prek: https://wiki.apache.org/tomcat/FAQ/FDA_Validation (23. januar 2017).
18. Kovačič, Andrej in Mirko Vintar. 1994. *Načrtovanje in gradnja informacijskih sistemov*. Ljubljana: DZS.
19. Lieberman, Gregory in Magdalena R. Naylor. 2012. Interactive voice response technology for symptom monitoring and as an adjunct to the treatment of chronic pain. *Translational Behavioral Medicine* 2 (1): 93–101.
20. Llanos, Adana A.M., Jesica L. Krok, Juan Peng, Michael L. Pennell, Mara Z. Vitolins, Cecilia R. Degraffinreid in Electra D. Pasket. 2014. Effects of a walking intervention using mobile technology and interactive voice response on serum adipokines among postmenopausal women at increased breast cancer risk. *Horm Cancer* 5 (2): 98–103.
21. Maheu, Marlene M., Pamela Whitten in Ace Allen. 2001. *E-Health, Telehealth, and Telemedicine: A Guide to Startup and Success*. San Francisco: Jossey-Bass Inc.
22. Midanik, Lorraine T. in Thomas K. Greenfield. 2008. Interactive Voice Response Versus Computer-Assisted Telephone Interviewing (CATI) Surveys and Sensitive Questions: The 2005 National Alcohol Survey. *Journal of Studies on Alcohol and Drugs* 69 (4): 580–588.
23. *Ministrstvo za zdravje*. Dostopno prek: <http://www.mz.gov.si> (17. januar 2017).
24. Mundt, James C., Dayna S Geralts in Heidi K. Moore. 2006. Dial "T" for Testing: Technological Flexibility in Neuropsychological Assessment. *Telemedicine and e-Health* 12 (3): 317–323.
25. *MySQL Developers*. Dostopno prek: <https://dev.mysql.com> (23. januar 2017).
26. Olufemi, Moses, Charles Uwadia in Adio Taofeek Akinwale. 2009. A framework for the design of speech-enabled self-care e-health systems. *The 9th International Conference on Electronic Business, Macau*. Dostopno prek: <http://iceb.nccu.edu.tw/proceedings/2009/49-55.pdf> (24. februar 2016).
27. *Oracle*. Dostopno prek: <http://www.oracle.com> (17. januar 2017).
28. *Oracle Docs*. Dostopno prek: <https://docs.oracle.com> (3. januar 2017).

29. Petek-Šter, Marija. 2014. *Zvišan krvni tlak: Navodila za bolnike*. Dostopno prek: <http://www.drmed.org/wp-content/uploads/2014/06/IV-261.pdf> (29. april 2014).
30. *RIS*. Dostopno prek: <http://www.ris.org> (17. januar 2017).
31. Scansoft. 2005. *OSR Platform Integration Manual Fourth Edition, Update 2*. Burlington: ScanSoft.
32. *Spring Framework*. Dostopno prek: <https://projects.spring.io/spring-framework> (3. januar 2017).
33. Strouhal, Peter. 2014. *Sladkorna bolezen: Navodila za bolnike*. Dostopno prek: <http://www.drmed.org/wp-content/uploads/2014/06/IX-84.pdf> (29. april 2016).
34. *Telekom Slovenije*. Dostopno prek: <http://www.telekom.si> (26. december 2016).
35. *Thymeleaf*. Dostopno prek: <http://www.thymeleaf.org> (22. januar 2017).
36. Vester, Irena. 2014. *Srčno popuščanje: Navodila za bolnike*. Dostopno prek: <http://www.drmed.org/wp-content/uploads/2014/06/IV-251.pdf> (29. april 2016).
37. *W3C*. Dostopno prek: <http://www.w3.org> (21. junij 2016).
38. Zupan, Gregor. 2015. *Uporaba interneta v gospodinjstvih in pri posameznikih, Slovenija, 2015*. SURS. Dostopno prek: <http://www.stat.si/StatWeb/prikazinovico?id=5509&idp=10&headerbar=8> (17. januar 2017).

PRILOGE

Priloga A: Izvorna koda projekta telezdravje.si

Izvorna koda je objavljena na portalu GitHub in je javno dostopna. Ker sem pri razvoju uporabljal angleški jezik, je tudi ime projekta »telehealth« angleško. Izvorna koda je na voljo na naslovu <https://github.com/bvukic/telehealth>.

Priloga B: Speech Recognition Grammar

Datoteke »speech recognition grammar« so sestavni del izvorne kode, ki je v celoti objavljena in dostopna (glej prilogo A). Spodaj je naveden najenostavnejši (boolean.grxml) in najkompleksnejši (numeric.grxml) primer uporabe jezika »speech recognition grammar« za vnos podatkov z uporabo tehnologije ASR.

Datoteka boolean.grxml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project
Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->

<!DOCTYPE grammar PUBLIC "-//W3C//DTD GRAMMAR 1.0//EN"
    "http://www.w3.org/TR/speech-grammar/grammar.dtd">

<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/06/grammar
http://www.w3.org/TR/speech-grammar/grammar.xsd"
    tag-format="swi-semantics/1.0"
    version="1.0"
    xml:lang="sl-SI"
    mode="voice"
    root="ROOT">

    <!-- <lexicon
uri="http://www.bosti.org/telehealth/static/ivr/grammar/numbers.xml"/> -->
    <meta name="swirec_simple_result_key" content="result"/>

    <rule id="ROOT" scope="public">
        <tag>result = 0;</tag>
        <one-of>
            <item>ne<tag>result = 0;</tag></item>
            <item>da<tag>result = 1;</tag></item>
            <item>ja<tag>result = 1;</tag></item>
        </one-of>
    </rule>

</grammar>
```

Datoteka numeric.grxml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project
Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->

<!DOCTYPE grammar PUBLIC "-//W3C//DTD GRAMMAR 1.0//EN"
    "http://www.w3.org/TR/speech-grammar/grammar.dtd">

<grammar xmlns="http://www.w3.org/2001/06/grammar"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2001/06/grammar
http://www.w3.org/TR/speech-grammar/grammar.xsd"
    tag-format="swi-semantics/1.0"
    version="1.0"
    xml:lang="sl-SI"
    mode="voice"
    root="ROOT">

    <!-- <lexicon
uri="http://www.bosti.org/telehealth/static/ivr/grammar/numbers.xml"/> -->
    <meta name="swirec_simple_result_key" content="result"/>

    <rule id="ROOT" scope="public">
        <tag>result = 0.0;</tag>
        <one-of>
            <item>
                <item repeat="0-1">
                    <ruleref uri="#kilo"/>
                    <tag>
                        result = result + kilo.value;
                    </tag>
                </item>
                <item repeat="0-1">
                    <ruleref uri="#hecto"/>
                    <tag>
                        result = result + hecto.value;
                    </tag>
                </item>
                <item repeat="0-1">
                    <ruleref uri="#num1_99"/>
                    <tag>
                        result = result + num1_99.value;
                    </tag>
                </item>
                <item repeat="0-1">
                    <ruleref uri="#fraction"/>
                    <tag>
                        result = result + fraction.value;
                    </tag>
                </item>
            </item>
        </one-of>
    </rule>
</grammar>
```

```

        <item>
            <ruleref uri="#num0"/>
        </item>
        <item repeat="0-1">
            <ruleref uri="#fraction"/>
            <tag>
                result = result + fraction.value;
            </tag>
        </item>
    </one-of>
</rule>

<rule id="fraction">
    <tag>value = 0.0;</tag>
    <item>
        <one-of>
            <item>celih</item>
            <item>cele</item>
            <item>cela</item>
        </one-of>
    </item>
    <item>
        <one-of>
            <item>
                <ruleref uri="#num0"/> <ruleref uri="#num1_9"/>
                <tag>value = num1_9.value / 100;</tag>
            </item>
            <item>
                <ruleref uri="#num1_99"/>
                <tag>
                    if (num1_99.value &lt; 10) {
                        value = num1_99.value / 10;
                    } else {
                        value = num1_99.value / 100;
                    }
                </tag>
            </item>
        </one-of>
    </item>
</rule>

<rule id="num0">
    <tag>value = 0.0;</tag>
    <one-of>
        <item>nič</item>
        <item>nula</item>
    </one-of>
</rule>

<rule id="kilo">
    <tag>value = 0.0;</tag>
    <one-of>
        <item>tisoč<tag>value = 1000.0;</tag></item>
        <item>dva tisoč<tag>value = 2000.0;</tag></item>
        <item>tri tisoč<tag>value = 3000.0;</tag></item>
        <item>štiri tisoč<tag>value = 4000.0;</tag></item>
        <item>pet tisoč<tag>value = 5000.0;</tag></item>
        <item>šest tisoč<tag>value = 6000.0;</tag></item>
    </one-of>
</rule>

```

```

        <item>sedem tisoč<tag>value = 7000.0;</tag></item>
        <item>osem tisoč<tag>value = 8000.0;</tag></item>
        <item>devet tisoč<tag>value = 9000.0;</tag></item>
    </one-of>
</rule>

<rule id="hecto">
    <tag>value = 0.0;</tag>
    <one-of>
        <item>sto<tag>value = 100.0;</tag></item>
        <item>dve sto<tag>value = 200.0;</tag></item>
        <item>tri sto<tag>value = 300.0;</tag></item>
        <item>štiri sto<tag>value = 400.0;</tag></item>
        <item>pet sto<tag>value = 500.0;</tag></item>
        <item>šest sto<tag>value = 600.0;</tag></item>
        <item>sedem sto<tag>value = 700.0;</tag></item>
        <item>osem sto<tag>value = 800.0;</tag></item>
        <item>devet sto<tag>value = 900.0;</tag></item>
    </one-of>
</rule>

<rule id="num1_9">
    <tag>value = 0.0;</tag>
    <one-of>
        <item>ena<tag>value = 1.0</tag>
        </item>
        <item>
            <one-of>
                <item>dva</item>
                <item>dve</item>
            </one-of>
            <tag>value = 2.0</tag>
        </item>
        <item>tri<tag>value = 3.0</tag></item>
        <item>štiri<tag>value = 4.0</tag></item>
        <item>pet<tag>value = 5.0</tag></item>
        <item>šest<tag>value = 6.0</tag></item>
        <item>sedem<tag>value = 7.0</tag></item>
        <item>osem<tag>value = 8.0</tag></item>
        <item>devet<tag>value = 9.0</tag></item>
    </one-of>
</rule>

<rule id="num10_19">
    <tag>value = 0.0;</tag>
    <one-of>
        <item>deset<tag>value = 10.0</tag></item>
        <item>enajst<tag>value = 11.0</tag></item>
        <item>dvanajst<tag>value = 12.0</tag></item>
        <item>trinajst<tag>value = 13.0</tag></item>
        <item>štirinajst<tag>value = 14.0</tag></item>
        <item>petnajst<tag>value = 15.0</tag></item>
        <item>šestnajst<tag>value = 16.0</tag></item>
        <item>sedemnajst<tag>value = 17.0</tag></item>
        <item>osemnajst<tag>value = 18.0</tag></item>
        <item>devetnajst<tag>value = 19.0</tag></item>
    </one-of>
</rule>

```

```

    <rule id="num20_99">
      <tag>value = 0.0;</tag>
      <item repeat="0-1"><ruleref uri="#num1_9"/>in<tag>value =
num1_9.value</tag></item>
      <item><ruleref uri="#deca"/><tag>value = value +
deca.value;</tag></item>
    </rule>

    <rule id="deca">
      <tag>value = 0.0;</tag>
      <one-of>
        <item>dvajset<tag>value = 20.0;</tag></item>
        <item>trideset<tag>value = 30.0;</tag></item>
        <item>štirideset<tag>value = 40.0;</tag></item>
        <item>petdeset<tag>value = 50.0;</tag></item>
        <item>šestdeset<tag>value = 60.0;</tag></item>
        <item>sedemdeset<tag>value = 70.0;</tag></item>
        <item>osemdeset<tag>value = 80.0;</tag></item>
        <item>devetdeset<tag>value = 90.0;</tag></item>
      </one-of>
    </rule>

    <rule id="num1_99">
      <tag>value = 0.0;</tag>
      <one-of>
        <item><ruleref uri="#num1_9"/><tag>value =
num1_9.value;</tag></item>
        <item><ruleref uri="#num10_19"/><tag>value =
num10_19.value;</tag></item>
        <item><ruleref uri="#num20_99"/><tag>value =
num20_99.value;</tag></item>
      </one-of>
    </rule>

</grammar>

```

Priloga C: Call Data Record

Spodaj priloženi Call Data Record je dnevniški zapis klica s platforme HP OpenCall Media Platform in prikazuje dnevniško sled klica na glasovni uporabniški vmesnik. Iz dokumenta so razvidni vnosi podatkov z uporabo prepoznavalnika govora.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<CDR asr="asr.asr://asr" callid="vxi_dialog_37_28_5505191" level="3"
  localuri="+38651951155" redirect=""
  remoteuri="sip:+386*****@IMS.MOBITEL.SI;user=phone"
  serviceid="VukicB_Telehealth"
  sessionid="http://10.122.12.37/ccxmlsession_37_28_13561074"
  starttime="2017-01-28T19:38:04.333" stoptime="2017-01-28T19:39:39.379"
  url="http://telezdravje.si/telehealth/api/vxml/login" version="1.0"
  voicexmlinterpreter="bpl-ocmp1.ts.telekom.si"
  xmlns="http://www.hp.com/ocmp/2004/03/cdr"
  xmlns:debug="http://www.hp.com/ocmp/2004/03/cdr-debug"
  xmlns:log="http://www.hp.com/ocmp/2004/03/cdr-log">
  <goto msec="1">http://telezdravje.si/telehealth/api/vxml/login</goto>
  <dialogstart msec="175"/>
  <log:info msec="176">Logging in user vukicb </log:info>

```

```

    <goto
msec="176">http://telezdravje.si/telehealth/j_spring_security_ivr_login?j_u
sername=vukicb</goto>
    <goto
msec="315">http://telezdravje.si/telehealth/api/vxml/input/load/0</goto>
    <log:info msec="567"/>
    <play
msec="676">http://telezdravje.si/telehealth/static/ivr/media/system/welcome
.wav</play>
    <play
msec="3365">http://telezdravje.si/telehealth/static/ivr/media/system/input_
data_notification.wav</play>
    <play
msec="8625">http://telezdravje.si/telehealth/static/ivr/media/medical/input
_data_gloukoze.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="14621"
results="1"
        score="89" threshold="50">&#x161;est celih dva<result>
            <interpretation confidence="89"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/numer
ic.grxml">
                <input mode="speech">&#x161;est celih dva</input>
                <instance>6.2</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="14622">Recognized input value: 6.2 </log:info>
    <goto
msec="14622">http://telezdravje.si/telehealth/api/vxml/say/NUMERIC?value=6.
2</goto>
    <play
msec="14776">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="15855">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/6.wav</play>
    <play
msec="16265">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/point.wav</play>
    <play
msec="17025">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/2.wav</play>
    <play msec="17495">builtin:break/500</play>
    <play
msec="18015">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="21471"
results="1"
        score="96" threshold="50">da<result>
            <interpretation confidence="96"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
                <input mode="speech">da</input>
                <instance>1</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="21472">Recognized input: 1 </log:info>
    <log:info msec="21472">User confirmed value 6.2 </log:info>
    <log:info msec="21472">Storing value 6.2 </log:info>

```



```

    <goto
msec="21472">http://telezdravje.si/telehealth/api/vxml/input/store/0?value=
6.2</goto>
    <log:info msec="21861"/>
    <play
msec="21965">http://telezdravje.si/telehealth/static/ivr/media/medical/inpu
t_data_bodyweight.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="28677"
results="1"
        score="67" threshold="50">pet in devetdeset<result>
            <interpretation confidence="67"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/numer
ic.grxml">
                <input mode="speech">pet in devetdeset</input>
                <instance>95</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="28677">Recognized input value: 95 </log:info>
    <goto
msec="28678">http://telezdravje.si/telehealth/api/vxml/say/NUMERIC?value=95
</goto>
    <play
msec="28845">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="29775">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/5.wav</play>
    <play
msec="30275">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/and.wav</play>
    <play
msec="30955">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/90.wav</play>
    <play msec="32135">builtin:break/500</play>
    <play
msec="32655">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="36182"
results="1"
        score="60" threshold="50">potrdi<result>
            <interpretation confidence="60"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
                <input mode="speech">potrdi</input>
                <instance>1</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="36183">Recognized input: 1 </log:info>
    <log:info msec="36183">User confirmed value 95 </log:info>
    <log:info msec="36183">Storing value 95 </log:info>
    <goto
msec="36183">http://telezdravje.si/telehealth/api/vxml/input/store/1?value=
95</goto>
    <log:info msec="36540"/>
    <play
msec="36642">http://telezdravje.si/telehealth/static/ivr/media/medical/inpu
t_data_spodnji_krvni_tlak.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="43490"
results="1"

```

```

    score="98" threshold="50">osem in osemdeset<result>
      <interpretation confidence="98"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/numer
ic.grxml">
        <input mode="speech">osem in osemdeset</input>
          <instance>88</instance>
        </interpretation>
      </result>
    </recognition>
    <log:info msec="43491">Recognized input value: 88 </log:info>
    <goto
msec="43491">http://telezdravje.si/telehealth/api/vxml/say/NUMERIC?value=88
</goto>
    <play
msec="43647">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="44595">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/8.wav</play>
    <play
msec="45215">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/and.wav</play>
    <play
msec="45805">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/80.wav</play>
    <play msec="46785">builtin:break/500</play>
    <play
msec="47305">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="50793"
results="1"
      score="89" threshold="50">da<result>
        <interpretation confidence="89"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
          <input mode="speech">da</input>
            <instance>1</instance>
          </interpretation>
        </result>
      </recognition>
    <log:info msec="50793">Recognized input: 1 </log:info>
    <log:info msec="50793">User confirmed value 88 </log:info>
    <log:info msec="50793">Storing value 88 </log:info>
    <goto
msec="50793">http://telezdravje.si/telehealth/api/vxml/input/store/2?value=
88</goto>
    <log:info msec="51170"/>
    <play
msec="51273">http://telezdravje.si/telehealth/static/ivr/media/medical/inpu
t_data_zgornji_krvni_tlak.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="57404"
results="1"
      score="86" threshold="50">sto tri in dvajset<result>
        <interpretation confidence="86"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/numer
ic.grxml">
          <input mode="speech">sto tri in dvajset</input>
            <instance>123</instance>
          </interpretation>
        </result>
      </recognition>

```

```

    <log:info msec="57404">Recognized input value: 123 </log:info>
    <goto
msec="57404">http://telezdravje.si/telehealth/api/vxml/say/NUMERIC?value=12
3</goto>
    <play
msec="57562">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="58505">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/100.wav</play>
    <play
msec="59175">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/3.wav</play>
    <play
msec="59595">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/and.wav</play>
    <play
msec="60185">http://telezdravje.si/telehealth/static/ivr/media/say//numeric
/20.wav</play>
    <play msec="61075">builtin:break/500</play>
    <play
msec="61595">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="65207"
results="1"
    score="94" threshold="50">ja<result>
    <interpretation confidence="94"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
    <input mode="speech">ja</input>
    <instance>1</instance>
    </interpretation>
    </result>
</recognition>
    <log:info msec="65208">Recognized input: 1 </log:info>
    <log:info msec="65208">User confirmed value 123 </log:info>
    <log:info msec="65208">Storing value 123 </log:info>
    <goto
msec="65208">http://telezdravje.si/telehealth/api/vxml/input/store/3?value=
123</goto>
    <log:info msec="65561"/>
    <play
msec="65663">http://telezdravje.si/telehealth/static/ivr/media/medical/inpu
t_data_mood.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="70368"
results="1"
    score="57" threshold="50">dobro<result>
    <interpretation confidence="57"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/condi
tion.grxml">
    <input mode="speech">dobro</input>
    <instance>3</instance>
    </interpretation>
    </result>
</recognition>
    <log:info msec="70369">Recognized input value: 3 </log:info>
    <goto
msec="70369">http://telezdravje.si/telehealth/api/vxml/say/SCALE_CONDITION?
value=3</goto>

```

```

    <play
msec="70521">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="71465">http://telezdravje.si/telehealth/static/ivr/media/say//conditi
on/3.wav</play>
    <play msec="72195">builtin:break/500</play>
    <play
msec="72715">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="75975"
results="1"
        score="96" threshold="50">ja<result>
            <interpretation confidence="96"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
                <input mode="speech">ja</input>
                <instance>1</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="75976">Recognized input: 1 </log:info>
    <log:info msec="75976">User confirmed value 3 </log:info>
    <log:info msec="75976">Storing value 3 </log:info>
    <goto
msec="75976">http://telezdravje.si/telehealth/api/vxml/input/store/4?value=
3</goto>
    <log:info msec="76330"/>
    <play
msec="76430">http://telezdravje.si/telehealth/static/ivr/media/medical/inpu
t_data_gibanje.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="81726"
results="1"
        score="94" threshold="50">malo<result>
            <interpretation confidence="94"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/quant
ity.grxml">
                <input mode="speech">malo</input>
                <instance>1</instance>
            </interpretation>
        </result>
    </recognition>
    <log:info msec="81727">Recognized input value: 1 </log:info>
    <goto
msec="81727">http://telezdravje.si/telehealth/api/vxml/say/SCALE_QUANTITY?v
alue=1</goto>
    <play
msec="81878">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_pre.wav</play>
    <play
msec="82825">http://telezdravje.si/telehealth/static/ivr/media/say//quantit
y/1.wav</play>
    <play msec="83585">builtin:break/500</play>
    <play
msec="84105">http://telezdravje.si/telehealth/static/ivr/media/system/input
_data_confirm_post.wav</play>
    <recognition cause="success" language="sl-SL" log="" msec="87730"
results="1"
        score="96" threshold="50">ja<result>

```

```

        <interpretation confidence="96"
grammar="http://telezdravje.si:80/telehealth/static/ivr/grammar/voice/confi
rm.grxml">
        <input mode="speech">ja</input>
        <instance>1</instance>
    </interpretation>
    </result>
</recognition>
<log:info msec="87731">Recognized input: 1 </log:info>
<log:info msec="87731">User confirmed value 1 </log:info>
<log:info msec="87731">Storing value 1 </log:info>
<goto
msec="87731">http://telezdravje.si/telehealth/api/vxml/input/store/5?value=
1</goto>
    <log:info msec="87890">Data input complete</log:info>
    <play
msec="87890">http://telezdravje.si/telehealth/static/ivr/media/system/input
_complete.wav</play>
    <play
msec="90665">http://telezdravje.si/telehealth/static/ivr/media/system/goodb
ye.wav</play>
        <vxmlevent event="connection.disconnect.hangup" msec="94995"/>
        <log:info msec="94995">User disconnected. Logging out...</log:info>
    <goto
msec="94995">http://telezdravje.si/telehealth/api/vxml/logout</goto>
        <log:info msec="95046">User vukicb logged out.</log:info>
</CDR>

```