

**RAZVOJ RESTFUL INTERNET SERVIS ZA DALJINSKO UPRAVLJANJE
PAMETNOM KUĆOM****DEVELOPMENT OF THE RESTFUL WEB SERVICE FOR REMOTE SMART HOME
CONTROL**

Stefan Nikolić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – U radu je prikazana internet RESTful aplikacija koja omogućuje korisniku upravljanje pametnom kućom preko interneta. Takođe, opisana je i Android aplikacija koja koristi resurse servisa kao i komunikacija između servisa i glavnog kućnog kontrolera.

Ključne reči: Internet aplikacija, RESTful, Android, Raspberry Pi, Baza podataka

Abstract – This paper represents development of RESTful web application which provides control over the smart house via the internet. It also describes Android application which consumes service resources and the communication between the web application and the main smart home controller.

Keywords: Web application, RESTful, Android, Raspberry Pi, Data base

1. UVOD

Prve pametne kuće su bile ideje, a ne građevine. Decenijama je naučna fantastika istraživala ideju o kućnoj automatizaciji. Pisci poput Rej Bredburija su izmišljali budućnost u kojoj su kuće interaktivne, svesne svog postojanja i samoupravljujuće. U jednoj od svojih priča Rej opisuje svet u kojem je ljudska vrsta izumrla, ali su kuće nastavile da funkcionišu [1].

Iako je popularnost pametnih kuća porasla u poslednje dve decenije, one postoje već duže vreme. U početku su to bile male hardverske aplikacije koje su dovele do razvoja pametnih kuća kakve poznajemo danas. Početkom 20. veka, nastaju prvi kućni uređaji poput usisivača. Iako ih u današnjem smislu ne bi nazvali pametnim uređajem, oni su ipak bili velika tehnološka inovacija za to vreme. Ubrzo nakon usisivača napravljen je i prvi frižider, a potom i mašine za pranje i sušenje veša, kao i mnoge druge.

Trenutno u eri virtuelnih svetova, sajber ratova i društvenih mreža, gde se sve lične i poslovne informacije skladište na udaljene servise, ne postoji ništa što bi sprečilo ljude da i svojim kućava pristupaju i upravljaju preko interneta. Ovaj rad se bavi razvojem jedne takve aplikacije koja bi omogućila korisnicima pristup pametnoj

kući preko internet pretraživača, ili putem android aplikacije.

2. SISTEM PAMETNE KUĆE

Automatizacija se može definisati kao upotreba kontrolisanih sistema i informacionih tehnologija za smanjenje čovekovog angažovanja u proizvodnji dobara i usluga. Koliko god tehnologija koju koristi prosečan čovek u svom životnom okruženju izgleda zastarela, ona u većini slučajeva koristi neki vid automatizacije. Tipični primeri automatizacije iz svakodnevnog života su tajmeri u aparatima za kafu, daljinsko upravljanje i automatsko uključivanje svetla u autu prilikom otvaranja vrata.

Kada se govori o kućnoj automatizaciji i pametnim kućama, mogućnosti u automatizaciji idu korak dalje. Umesto pojedinačnih uređaja koji rade nezavisno jedni od drugih, pametne kuće integrišu više podsistema kojima se upravlja pomoću jednog glavnog kućnog kontrolera. Ova centralna upravljajuća jedinica predstavlja kičmu kućne automatizacije koja prima ulazne podatke od ostalih kućnih uređaja, izdaje naredbe i upravlja svime.

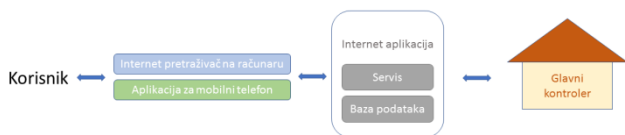
Uobičajeni elementi kućne automatizacije su:

- sistemi za kontrolu grejanja, klime i ventilacije,
- bela tehnika i manji namenski kućni aparati (mašine za sušenje veša, mašine za pranje veša, refrižideri, zamrzivači, mašine sa pranje suđa, rerne, aparati za kafu, mikrotalasne rerne, itd.),
- potrošačka elektronika (televizori, radio, stereo sistemi, konzole za igrice, itd.),
- stolarija (vrata, prozori, roletne, itd.).

Komunikacija između korisnika i glavnog kontrolera se može realizovati pomoću internet servisne aplikacije. Servisna aplikacija bi se nalazila na udaljenom serveru i sadržala bi svu biznis logiku. Sa računara bi joj se pristupalo pomoću internet pretraživača, dok za mobilne telefone bi se napravile jednostavne aplikacije koje predstavljaju produžetak interfejsa internet aplikacije, slika 1. Na ovaj način je rešen problem portabilnosti, jer sistem ne zavisi od operativnog sistema računara korisnika. Takođe, korisnik nema odgovornosti oko instaliranja programa i njegovog održavanja, a rešen je i problem distribucije softvera. Korisnik verovatno neće primetiti razliku u funkcionisanju aplikacije prilikom uključivanja novih funkcionalnosti ili rešenja postojećih problema. Ovim putem bi se napravila univerzalna platforma za sve korisnike pametnih kuća i unapredio bi se kvalitet usluga.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Gordana Ostojić, red. prof.



Slika 1. Komunikacija između korisnika i pametne kuće

3. INTERNET APLIKACIJA

U računarstvu, internet aplikacija je klijentsko-serverski računarski program koji se pokreće u internet pretraživaču. Tipične internet aplikacije su webmail, internet stranice za kupovinu, internet stranice za aukcije, internet enciklopedije, društvene mreže, internet stranice za slušanje muzike i gledanje videa, i mnoge druge. Generalna razlika između dinamične internet stranice i internet aplikacije je nejasna. Internet stranice koje imaju slične funkcionalnosti poput aplikacija za personalne računare se najčešće nazivaju internet aplikacijama, a povremeno i mobilnim aplikacijama ukoliko su prilagođene prikazivanju na malim ekranima.

Struktura internet aplikacije se obično deli na slojeve zavisno od njene kompleksnosti. Tradicionalno su se sastajale od jednog sloja, korisničkog interfejsa, ali zbog proširenja funkcionalnosti vremenom se stvorila potreba sa n-slojevitom strukturom. Najčešće su troslojne aplikacije gde srednji sloj sadrži biznis logiku, a treći sloj predstavlja bazu podataka.

3.1. Baza podataka

Baza podataka je organizovana kolekcija informacija koje se čuvaju u elektronskim uređajima. Dizajneri formiraju baze podataka tako da one što približnije opisuju realnost na način koji podržava procesuiranje podataka. Primer može biti modelovanje slobodnih soba u hotelima tako da podržava pretragu hotela sa slobodnom sobom određenog tipa.

Kompleksnost baze podataka uglavnom zavisi od zadatka koji treba da ispuni. U ovom radu korišćena je jednostavna baza podataka koja se sastoji iz dve tabele:

- tabela korisnika,
- tabela uređaja.

Tabela korisnika se sastoji iz tri kolone:

- korisničko ime,
- šifra korisnika,
- IP adresa pametne kuće,
- ime glavnog kontrolera, i
- šifra glavnog kontrolera.

Korisničko ime uz šifru služi za verifikaciju korisnika u internet i android aplikaciji što znači da ne smeju postojati dva korisnika sa istim korisničkim imenom, dok ime i šifra kontrolera služe za verifikaciju korisnika prilikom slanja komandi kontroleru sa servera.

Tabela uređaja se sastoji iz deset kolona:

- identifikacioni niz znakova,
- ime uređaja,
- IP adresa pametne kuće,

- lokalni identifikacioni niz znakova,
- smer,
- tip vrednosti,
- trenutna vrednost,
- zahtevana vrednost,
- maksimalna vrednost, i
- minimalna vrednost

Tabela uređaja sadrži uređaje svih korisnika i zbog toga tabela sadrži kolonu sa IP adresom glavnog kontrolera kako bi se napravila povezanost između uređaja i korisnika, tj. pametne kuće. Identifikacioni niz znakova je jedinstven za svaki uređaj i nastaje spajanjem IP adrese glavnog kontrolera na koji je uređaj povezan i lokalnog identifikacionog niza znakova. Lokalni identifikacioni niz znakova služi kako bi glavni kontroler mogao prepoznati za koji uređaj je poslata komanda. Taj niz znakova na primer može biti port i pin preko kojeg kontroler komunicira sa uređajem.

3.2. RESTful servis

Pojam internet servisa u širem smislu podrazumeva pružanje usluga jednom električnom uređaju od strane drugog električnog uređaja, koja komuniciraju preko WWW (World Wide Web). U užem smislu, internet servis podrazumeva servis implementiran u određenoj tehnologiji ili brendu, W3C Web of Service [2].

REST (Representational State Transfer) je stil softverske arhitekture namenjen distribuiranim hipermedijalnim sistemim kao WWW. REST specifikira određena ograničenja [3]:

- način identifikovanja resursa,
- uniformisani interfejs (GET, PUT, DELETE, POST i druge predefinisane CRUD HTTP metode),
- samo-opisive poruke,,
- stanje aplikacija upravljano hipermedijom (URL, Uniform Resource Locator, linkovi predstavljaju krajnje pristupne tačke pomoću kojih se manipuliše resursima), i
- Stateless stanje (ne zavisi od tehnologija koje se koriste).

Primena ovih ograničenja na internet servise pojačava njihove pozitivne osobine kao što su performans, skalabilnost, izmenljivost i druge. U REST arhitekturi, podaci i funkcionalnosti se posmatraju kao resursi i pristupa im se putem URI (Uniform Resource Identifiers) identifikatora, tipično kao URL na internetu. Pozivanje usluge, tj. metode, servisa preko URL-a bi izgledalo kao `https://internet_aplikacija/restful_servis/ime_metode/ulazni_parametar1/ulazni_parametar2`.

3.3. Interfejs

HTML (Hypertext Modeling Language) je standardni opisni jezik za kreiranje internet stranica i internet aplikacija. Uz CSS (Cascading Style Sheets) i JavaScript formira trijadu tehnologija koje čine jezgro WWW-a. Internet pretraživač preuzima HTML dototeke od internet servisa ili iz lokalne memorije i renderuje datoteke u multimedijalne internet stranice. HTML semantički opisuje strukturu internet stranice pomoću HTML

elementa (tagova). Standard prepoznaje predefinisane elemente koji definišu različite delove dokumenta poput naslova, paragrafa, liste, tabele, itd., kao i tagovi za meta podatke koji detaljnije opisuju sam dokument, poput kratkog opisa dokumenta, ključnih reči, podaci o autoru i slično.

U ovom radu internet stranicu čine pet HTML formi:

- login.html,
- index.html (slika 2),
- AddDigitalDevice.html,
- AddAnalogDevice.html, i
- DeleteDevice.html.



Slika 2. *index.html* forma

Kako HTML definiše strukturu i sadržaj stranice tako CSS određuje kako će sadržaj biti prikazan. CSS može definisati boju, veličinu, pozadinu, granice i mnoge druge attribute HTML elemenata, takođe dozvoljava nasleđivanje i prenošenje osobina između HTML elemenata. CSS je dizajniran sa namereom da se razdvoje način prezentovanja i sadržaj, dok se pre pojave CSS-a sve nalazilo u HTML formi. Odvajanje načina prezentovanja od sadržaja pruža veću fleksibilnost i kontrolu nad određenim specifikacijama prezentovanja.

JavaScript je dinamičan, slabo tipiziran i interpretiran programski jezik visokog nivoa. JavaScript podržava objektno-orijentisani, imperativni i funkcionalni način programiranja. Sadrži biblioteke za rad sa tekstom, nizovima, datumima i regularnim izrazima, ali ne i ulazno/izlazne funkcionalnosti, kao što su povezivanje, skladištenje podataka ili funkcionalnosti, za šta se oslanja na okruženje u kome se izvršava.

U ovom radu JavaScript, uz biblioteke JQuery i AJAX [5], služi za komunikaciju interfejsa sa srednjim slojem aplikacije.

4. ANDROID APLIKACIJA

Android operativni sistem je trenutno najrasprostranjeniji operativni sistem za mobilne telefone. Zasnovan je na Linux jezgru i prilagođen je tako da se može koristiti na većini mobilnih uređaja, kao što su mobilni telefoni, tablet računari, laptop računari ali i mnogi drugi uređaji.

Proizvođači koji koriste Android mogu ugraditi svoje dodatke u Android i na taj način učiniti njihove proizvode drugačijim od drugih [4].

Aplikacija se sastoji od samo jedne aktivnosti i prilikom aktiviranja prikazuje prozor za prijavljivanje korisnika. Nakon uspešnog prijavljivanja korisnika, prozor za prijavljivanje se sakriva i otkriva se prozor sa listom uređaja (slika 3).



Slika 3. Prikaz uređaja u Android aplikaciji

5. SLANJE KOMANDI GLAVNOM KONTROLERU

U radu se koristi Raspberry Pi kao glavni kućni kontroler, koji je minijaturni računar veličine kreditne kartice. Povezivanje RESTful servisa sa Raspberry Pi-om vrši se preko SSH protokola (Secure Shell).

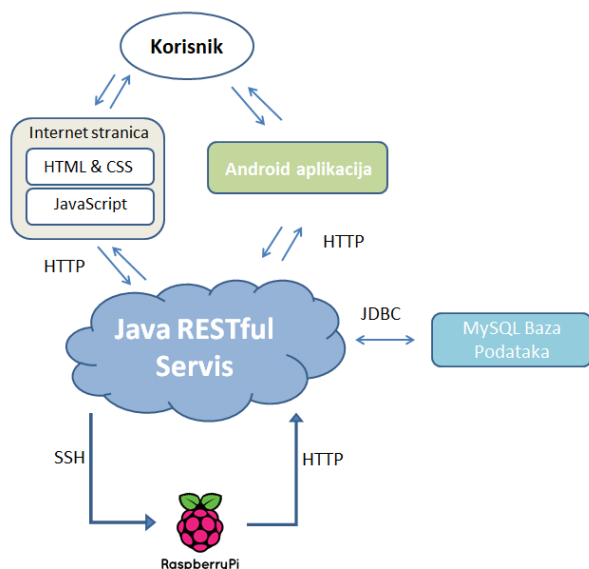
Raspbian operativni sistem ima u sebi već ugrađen SSH server pomoću kojeg je moguće konektovati se i kontrolisati Raspberry Pi. Nakon povezivanja sa SSH servisom, korisnik dobija pristup terminalu koji omogućava korisniku punu kontrolu nad sistemom. Pošto u ovom radu nije razvijan hardver pametne kuće, napravljena je Java aplikacija koja se poziva kroz terminal i simulira pozivanje resursa servisa za postavljanje trenutne vrednosti uređaja.

RESTful servis koristi JSch biblioteku da bi se konektovao na SSH servis [6] i prosleđuje komandu terminalu koja pokreće Java aplikaciju. Format komande je:

```
java -jar com.ftn.restapp.client.jar device_id requested_value
```

Tok informacija od korisnika do pametne kuće, i od pametne kuće do korisnika izgleda kao na slici 4. Korisnik menja stanje na interfejsu internet ili android aplikacije koje dalje putem HTTP protokola komuniciraju sa RESTful servisom. RESTful servis obrađuje podatke i komunicira sa bazom podata putem JDBC-a (Java DataBase Connectivity), a potom šalje komandu glavnom kontroleru putem SSH protokola.

Glavni kontroler vraća podatke servisu, koji menja stanje u bazi podataka i prikazuje povratnu informaciju korisniku kroz interfejs.



Slika 4. Tok informacije od korisnika do pametne kuće

6. SIGURNOST

Možda najvažnije pitanje koje utiče na popularnost i prihvatljivost ovog vida upravljanja pametnom kućom jeste pitanje sigurnosti. Da li je RESTful servis dovoljno bezbedan da bi mu se dala kontrola nad nečijim stambenim prostorom je verovatno prvo pitanje koje bi pojedinci postavili. Istina je da ne postoji idealna zaštita, ali velike društvene mreže na kojima se svakodnevno postavljaju velike količine ličnih podataka, poput Facebook-a [7], upravo koriste REST tip servisa.

Prvi stepen zaštite predstavlja kodiranje podataka. To se radi iz razloga što bilo ko, pod uslovom da poznaje resurse servisa, može čitati podatke iz baze samo koristeći URL u internet pretraživaču. Postoji veliki broj biblioteke koje se mogu koristiti u te svrhe i nije ih teško koristiti. Takođe postoje i alati za kodiranje URL linka.

Sa druge strane, strane kontrolera, postavlja se pitanje koliko je sigurna SSH komunikacija. U radu se za verifikaciju koriste samo ime i šifra, ali postoje i načini autorizacije pomoću enkriptovanih ključeva. U tom slučaju postoje dva ključa, privatni i javni, od kojih se jedan nalazi na kontroleru koji vrši verifikaciju pristupa, a drugi u sistemu koji pristupa kontroleru. Tada nema potrebe za verifikacijom imena i šifre jer je enkriptovan ključ dovoljna zaštita sama po sebi. Njih je skoro nemoguće hakovati, jedino ukoliko privatni ključ ne dođe u pogrešne ruke.

7. ZAKLJUČAK

U ovom radu je prikazan detaljan postupak izrade RESTful servisa, prikazan je način komunikacije sa servisom, i opsiana je komunikacija između servisa i glavnog kontrolera pametne kuće. Napravljene su internet i Android aplikacije koje omogućuju korisniku pristup servisu, napravljen je RESTful servis sa bazom podataka, i napravljena je Java aplikacija koja se nalazi u glavnom kontroleru i simulira odziv pametne kuće.

Ovakav sistem omogućava korisniku potpunu kontrolu nad stambenim prostorom sa bilo koje lokacije na planeti pod uslovom da ima pristup internetu. Mogućnosti pametne kuće su velike i njene usluge se mogu dalje unapređivati proširenjem baze podataka, da podaci budu specifičniji i da bolje reflektuju stvarne uređaje, proširenjem Android aplikacije sa svim opcijama internet aplikacije, unapređenjem korisničkog interfejsa i implementacijom visokog stepena zaštite podataka.

Baza podataka predstavlja bitan faktor u jednom ovakvom sistemu, jer od opštosti modela i podataka može zavisiti kompatibilnost sa stvarnim uređajima, dakle važno je da model što bolje reflektuje stvaran svet. Sa druge strane, kompleksnost baze podataka otežava njeno čitanje i komplikuje realizaciju servisa što može dovesti do problema u performansama sistema i kasnijeg održavanja sistema. Predlog za dalje unapređenje sistema je hijerarhijski model baze podataka gde se na vrhu nalazi tabela korisnika. Ispod tabele korisnika bila bi tabela pametnih kuća, jer jedan korisnik može imati više pametnih kuća. Potom, tabela uređaja koji čine pametne kuće, a ispod njih tabele senzora, aktuatora i prekidača od kojih se sastoje uređaji. Na dnu hijerarhije bi se nalazile tabele analognih i digitalnih vrednosti. Ovakvim modelom, gde komponente koje su niže u hijerarhiji opisuju komponente iznad njih, moguće je predstaviti gotovo sve od pojedinačnih uređaja do celokupnog sistema pametnih kuća.

8. LITERATURA

- [1] Raz Bradbury, "There Will Come Soft Rains", Collier, 1950.
- [2] www.w3.org (pristupljeno u septembru 2018.)
- [3] Leonard Richardson, Sam Ruby, "RESTful Web Services", O'Reilly, 2017.
- [4] James Steele, Nelson To, Android: izrada aplikacija pomoću paketa Android SDK, Beograd: Mikro knjiga, 2011.
- [5] jquery.com (pristupljeno septembru 2018)
- [6] www.jcraft.com (pristupljeno oktobru 2018)
- [7] developers.facebook.com (pristupljeno u oktobru 2018.)

Kratka biografija:



Stefan Nikolić rođen je u Nišu 1992. god. Posle završene osnovne škole upisuje gimnaziju "Stevan Sremac" u Nišu, smer prirodno-matematički. Srednju školu završava 2011. godine i iste godine upisuje Fakultet tehničkih nauka u Novom Sadu, odsek mehatronika, smer mehatronika, usmerenje mehatronika, robotika i automatizacija. Dobija zvanje diplomirani inženjer mehatronike 2015. godine