



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Норберт Немет

**Реализација корисничке апликације за
приступ функционалностима
послужиоца за аутоматску
конфигурацију TR-069 уређаја**

МАСТЕР РАД

Нови Сад, 2014.



УНИВЕРЗИТЕТ У НОВОМ САДУ ● ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6


КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска документација	
Тип записа, ТЗ:	Текстуални штампани материјал	
Врста рада, ВР:	Дипломски – мастер рад	
Аутор, АУ:	Норберт Немет	
Ментор, МН:	доц. др Милан Бјелица	
Наслов рада, НР:	Реализација корисничке апликације за приступ функционалностима послужоца за аутоматску конфигурацију TR-069 уређаја	
Језик публикације, ЈП:	Српски / латиница	
Језик извода, ЈИ:	Српски	
Земља публиковања, ЗП:	Република Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2014	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	7/53/16/3/43/0/0	
Научна област, НО:	Електротехника и рачунарство	
Научна дисциплина, НД:	Рачунарска техника	
Предметна одредница/Кључне речи, ПО:	Конфигурациони послужилац, корисничка апликација	
УДК		
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У овом раду је представљено једно решење графичке корисничке апликације за приступ функционалностима послужилаца који користи TR-069 протокол за комуникацију са корисничким уређајима.	
Датум прихватања теме, ДП:		
Датум одбране, ДО:		
Чланови комисије, КО:	Председник: проф. др Никола Теслић	
	Члан: проф. др Бранислав Тодоровић	Потпис ментора
	Члан, ментор: доц. др Милан Бјелица	



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Master Thesis
Author, AU :	Norbert Nemet
Mentor, MN :	Milan Bjelica, PhD
Title, TI :	Implementation of a user application for accessing TR-069 auto configuration server functionalities
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	2014
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	7/53/16/3/43/0/0
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	configuration server, user application
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	In this thesis a solution for a user application accessing TR-069 auto configuration server functionalities is presented.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: Nikola Teslić, PhD
	Member: Branislav Todorović, PhD
	Member, Mentor: Milan Bjelica, PhD
	Mentor's sign

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА МАСТЕР РАД	Датум:

(Податке уноси предметни наставник - ментор)

СТУДИЈСКИ ПРОГРАМ:	Рачунарство и аутоматика
РУКОВОДИЛАЦ СТУДИЈСКОГ ПРОГРАМА:	Никола Јорговановић

Студент:	Норберт Немет	Број	E2 25/2013
Област:	Пројектовање наменских рачунарских структура 2		
Ментор:	Милан Бјелица		
<p>НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:</p> <ul style="list-style-type: none"> - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; 			

НАСЛОВ МАСТЕР РАДА:

Реализација корисничке апликације за приступ функционалностима послужеоца за аутоматску конфигурацију TR-069 уређаја
--

ТЕКСТ ЗАДАТКА:

<p>У оквиру задатка је потребно реализовати корисничку апликацију за приступ функционалностима аутоконфигурационог послужеоца заснованог на TR-069 протоколу. У оквиру рада неопходно је прикупити захтеве на основу анализе TR-069 стандарда и спрежних подсистема, те на основу анализе постојећих решења. У оквиру спецификације решења апликације потребно је:</p> <ul style="list-style-type: none"> - дефинисати кориснике апликације; - дефинисати случајеве коришћења; - дефинисати структуру апликације; - одабрати погодне технологије реализације у односу на захтеве и на критеријум преносивости; - реализовати апликацију коришћењем апликативне спреге доступног TR-069 послужеоца; - верификовати решење у односу на захтеве и случајеве коришћења; - верификовати решење у односу на захтеве времена одзива, упоредном методом.

Руководилац студијског програма:	Ментор рада:

Примерак за: <input type="checkbox"/> - Студента; <input type="checkbox"/> - Ментора
--

Zahvalnost

Zahvaljujem se Milanu Bjelici na stručnoj pomoći, savetima i utrošenom vremenu.

Posebno se zahvaljujem rukovodstvu Instituta RT-RK na ukazanoj prilici da se bolje upoznam sa načinom rada u inženjerskom okruženju i budem uključen u proces razvoja novih programskih rešenja.

Zahvaljujem se roditeljima, Sandri i svima onima koji su na bilo koji način doprineli izradi ovog master rada.

SADRŽAJ

1. Uvod.....	1
2. Teorijske osnove	3
2.1 TR-069 Protokol.....	3
2.2 Prilagodni sloj poslužilaca za pristup funkcionalnostima	4
2.3 Mogući pristupi realizaciji korisničke aplikacije	5
2.3.1 Metode za razvoj Web aplikacija.....	6
2.4 Mehanizam za udaljeni poziv metoda poslužilaca u GWT aplikaciji	7
2.5 Postojeća rešenja	9
3. Koncept rešenja.....	11
3.1 Strukturni zahtevi i slučajevi korišćenja	11
3.2 Struktura Web aplikacije.....	15
4. Realizovano rešenje	19
4.1 Raspored modula.....	19
4.2 Izgled i funkcionalnosti realizovane aplikacije	22
4.2.1 Globalni izgled.....	22
4.2.2 Spisak uređaja	24
4.2.3 Pregled statusa uređaja	25
4.2.4 Detaljni pregled parametara.....	26
4.2.5 Stranica za osvežavanje programske podrške uređaja.....	27
4.2.6 Rukovanje grupama uređaja	28
4.2.7 Mapaza prikaz distribucije vrednosti parametra.....	29
4.2.8 Pozicije uređaja u nadgledanom području	29
4.2.9 Statistika sistema	30

4.2.10	Stranica za rukovanje nalogima vlasnika uređaja	31
4.2.11	Stranica za rukovanje korisničkim nalogima aplikacije	32
4.2.12	Stranica za rukovanje skriptovima.....	32
4.3	Specifična rešenja u okviru aplikacije.....	33
4.3.1	Osvežavanje modula aplikacije	33
4.3.2	Promenljiva stranica za praćenje statusa uređaja.....	35
5.	Evaluacija rešenja	37
5.1	Upoređivanje funkcionalnosti	37
5.2	Merenje vremena odziva	38
5.3	Potrošnja sistemskih resursa	39
6.	Zaključak	40
7.	Literatura.....	41

SPISAK SLIKA

Slika 1. Hijerarhija TR-069 protokola	3
Slika 2. Arhitektura visokog nivoa sistema baziranom na TR-069 protokolu	4
Slika 3. Autokonfiguracioni i nadzorni sistem.....	5
Slika 4. Dijagram realizacije GWT RPC	8
Slika 5. Redosled komunikacije sa poslužilacom	8
Slika 6. Korisnička aplikacija Motorola EDGE poslužioca.....	9
Slika 7. Korisnička aplikacija poslužioca kompanije Friendly Technologies	10
Slika 8. Koraci korisnika pri rešavanju problema.....	12
Slika 9. Koraci za nalaženje greške u programskoj podršci uređaja.....	12
Slika 10. Koraci za analizu stanja sistema	13
Slika 11. Koraci za registraciju novog korisnika	13
Slika 12. Koraci rada korisničke podrške pri rešavanju korisničkog problema.....	13
Slika 13. Slučajevi korišćenja aplikacije.....	14
Slika 14. Raspored modula u korisničkoj aplikaciji i veze između njih.	18
Slika 15. Dijagram klasa Web aplikacije	20
Slika 16. Postupak učitavanja nove stranice	21
Slika 17. Primer UiBinder deklarativnog opisa	22
Slika 18. Stranica za prijavljivanje.....	23
Slika 19. Opšti izgled aplikacije	23
Slika 20. Prozor aktivnosti	24
Slika 21. Spisak uređaja	24
Slika 22. Osnovnastranica za pregled statusa uređaja.....	25
Slika 23. Stanje za izmenu prikazanih parametra	26

Slika 24. Dijalog za dodavanje parametra.....	26
Slika 25. Stranica za detaljni pregled parametara	27
Slika 26. Izabrani parametar sa prikazanim promenama u vremenu	27
Slika 27. Stranica za osvežavanje programske podrške uređaja	28
Slika 28. Stranica za rukovanje grupama uređaja	28
Slika 29. Mapa sa kvalitetom DVB-T2 signala	29
Slika 30. Pozicije uređaja u nadgledanom području	30
Slika 31. Stranica za uvid u statistiku sistema	30
Slika 32. Stranica za rukovanje nalogima vlasnika uređaja	31
Slika 33. Stanje za izmenu podataka korisnika.....	31
Slika 34. Stranica za rukovanje korisničkim nalogima aplikacije.....	32
Slika 35. Stranica za rukovanje skriptovima.....	32
Slika 36. Dijalog za određivanje automatskog izvršenja skripte	33
Slika 37. Primer komunikacije između modula aplikacije i klase <i>StatusUpdater</i>	34
Slika 38. Stranica za praćenje statusa STB uređaja	35
Slika 39. Stranica za praćenje statusa pametnih utičnica i prekidača	35
Slika 40. Pokretanje druge stranice aplikacije	36
Slika 41. Pokretanje modula za praćenje statusa	36
Slika 42. Rezultati merenja vremena odziva.....	38
Slika 43. Rezultati merenjapotrošnje memorije	39

SPISAK TABELA

Tabela 1. Poređenje tipova aplikacija	6
Tabela 2. Analiza ispunjenosti strukturnih zahteva	37
Tabela 3. Poređenje funkcionalnosti	38

SKRAĆENICE

IPTV	- <i>Internet Protocol Television</i> , Televizija preko internet protokola
API	- <i>Application Programming Interface</i> , Sprega za razvoj programske podrške
GWT	- <i>Google Web Toolkit</i> , Guglov skup alata za razvoj Web aplikacija
RPC	- <i>Remote Procedure Call</i> , Udaljeni poziv metode
ACS	- <i>Autoconfiguration Server</i> , Poslužilac za automatsku konfiguraciju
CPE	- <i>Customer Premises Equipment</i> , Krajnji korisnički uređaj
CWMP	- <i>CPE WAN Management Protocol</i> , CPE WAN protokolupravljanja
HTTPS	- <i>HyperText Transfer Protocol Secure</i> , Osigurani protokol za prenos hiperteksta
NBI	- <i>Northbound Interface</i> , Sprežni sloj ACS-a i namenskih aplikacija
UI	- <i>User Interface</i> , Korisnička sprega
XML	- <i>Extensible Markup Language</i> , Proširiv opisni jezik
HTML	- <i>HyperText Markup Language</i> , Jezik za definisanje hiperteksta
STB	- <i>Set Top Box</i> , uređaj za prijem TV signala

1. Uvod

Broj uređaja potrošačke elektronike ubrzano raste u poslednjih par godina. Pored kvantitativnog rasta broja uređaja, dolazi do tehnološkog napretka u njihovom razvoju, čime je omogućeno još više funkcionalnosti za krajnje korisnike. Veći zahtevi korisničkih uređaja doprinose i sve većim zahtevima za fizičkim resursima. Multimedijalni uređaji, kao što su telefoni i televizori, već su uveliko zasnovani na internetu. Uređaji sa ovakvim potrebama zahtevaju od provajdera (eng. *Service provider*) komunikacionih i IPTV (eng. *Internet Protocol Television*) uslugada obezbedi besprekoran rad svojih mreža, kako bi se korisnicima omogućila što bolja usluga.

Razvoj mrežnih tehnologija i brzine razmene podataka je doveo do mogućnosti da se mreže sastavljene od uređaja potrošačke elektronike kontrolišu i nadgledaju sa udaljenih stanica. Poslužioc (eng. *Servers*) uređaja potrošačke elektronike su sistemi zasnovani na računaru, čiji je osnovni zadatak komunikacija sa uređajima sa jedne strane i manipulacija podacima sa druge strane.

Poslužioc upravljaju podacima tako što ih obrađuju, skladište, grupišu i isporučuju drugim aplikacijama ili korisnicima radi daljeg procesiranja ili grafičke prezentacije. Poslužioc za komunikaciju sa aplikacijama za naknadnu obradu ili grafičku predstavu komuniciraju preko svojih aplikativnih sprežnih slojeva (eng. *Application Programming Interface*, API). Protokoli koji se koriste u tim sprežnim slojevima su definisani od strane poslužilaca.

Da bi se omogućio pristup podacima i konfigurisanje uređaja, neophodne su grafičke aplikacije. Potrebno je da aplikacija bude dostupna sa udaljenih mesta radi olakšavanja rada tehničkog osoblja i da omogućuje efikasno rešenje problema osoblju korisničke podrške provajdera.

U ovom radu je predstavljeno jedno rešenje grafičke korisničke aplikacije za pristup funkcionalnostima poslužilaca koji koristi TR-069 protokol u komunikaciji sa korisničkim

uređajima. Aplikacija komunicira sa poslužilacem posredstvom GWT RPC (eng. *Google Web Toolkit Remote Procedure Call*) mehanizma i realizovana je u GWT (eng. *Google Web Toolkit*) okruženju. Aplikacija omogućuje nadgledanje mreže uređaja, pristupanje parametrima svakog uređaja, i određene napredne funkcionalnosti.

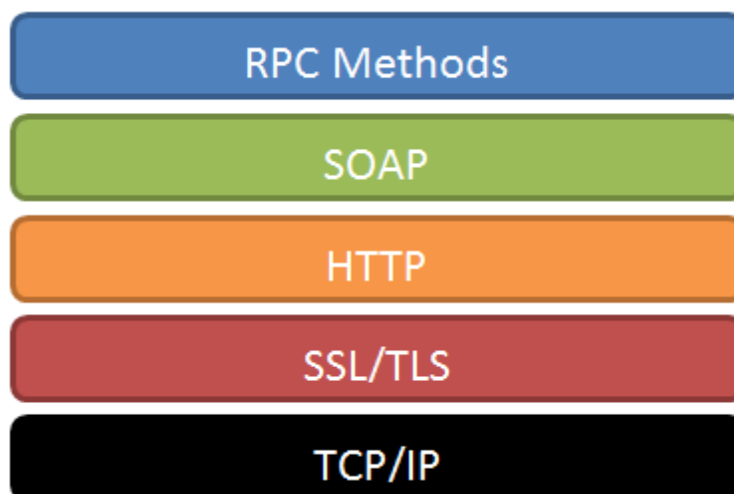
Rad je organizovan u sledećih nekoliko poglavlja:

1. **Teorijske** osnove – pokrivaju osnove protokola, postojeća rešenja i osnove okruženja korišćenog za razvoj.
2. **Koncept** rešenja – opisuje proces projektovanja aplikacije, korisničke slučajeve i grafički nacrt.
3. **Realizovano rešenje**– predstavlja programsku implementaciju aplikacije.
4. **Evaluacija** rešenja – predstavlja testiranje aplikacije sa poslužilacem u realnom okruženje i metričko poređenje sa sličnim rešenjima.
5. **Zaključak** – opisuje ispunjenost zahtevai budući rad.

2. Teorijske osnove

2.1 TR-069 Protokol

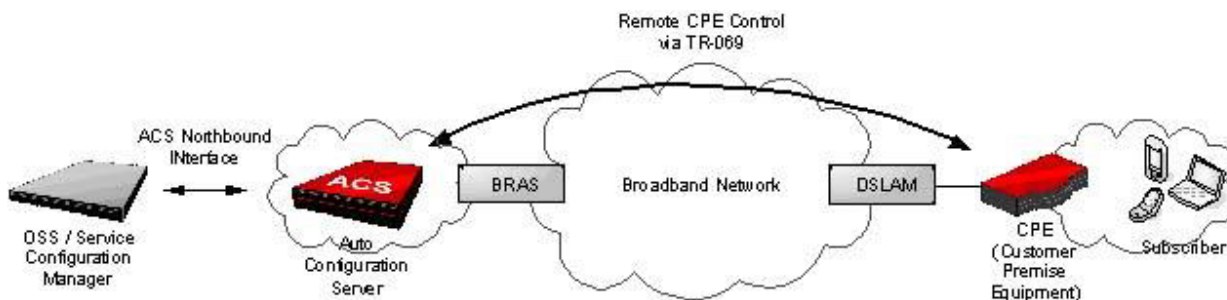
TR-069 je tehnička specifikacija protokola koji omogućuje daljinsko upravljanje i konfiguraciju korisničkih uređaja potrošačke elektronike. TR-069 protokol omogućava komunikaciju između entiteta u standardu. U standardu su propisani CPE (eng. *Customer Premises Equipment*) i ACS (eng. *Auto-Configuration Server*) uređaji. Prvi su korisnički uređaji kojima se pristupa preko centralne tačke – ACS servera. Za transport poruka između tačaka koriste se standardizovani načini i standardi. Protokole po nivoima prikazuje Slika 1.



Slika 1. Hijerarhija TR-069 protokola

Standard je zbog svoje fleksibilnosti postao najzastupljeniji standard ovog tipa. TR-069, poznat i kao CWMP (eng. *CPE Wan Management Protocol*), omogućuje prilagođavanje komunikacije prirodi korisničkih uređaja. Pomoću koncepta modela podataka moguće je prilagoditi komunikaciju i upravljanje parametrima mreže i korisničkih uređaja[1]. Upravo ovaj koncept je doprineo definisanju standardizovanih modela. Ovakva koncepcija standarda je

dovela do toga da nadležne institucije za razvijanje standarda u oblasti mrežnih tehnologija predlože TR-069 kao primarno rešenje za upravljanje i nadgledanje krajnjih uređaja. Arhitektura visokog nivoa sistema zasnovanog na TR-069 protokolu je prikazana na Slika 2.

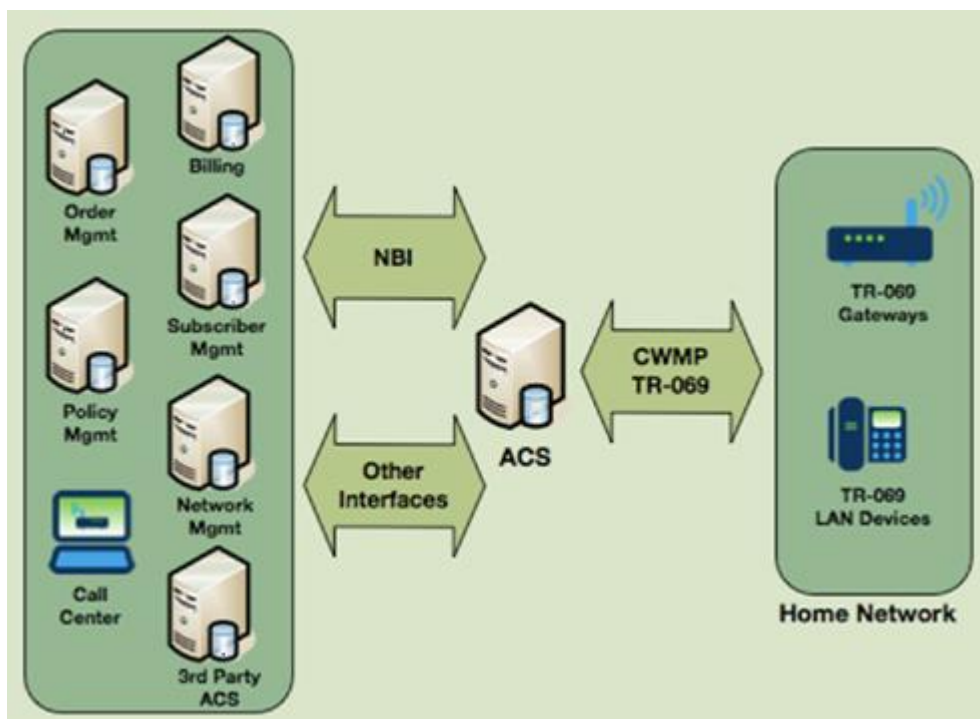


Slika 2. Arhitektura visokog nivoa sistema baziranom na TR-069 protokolu

Komunikacija između servera i korisničkih uređaja se odvija u sesijama (eng. *Session*). Sadržaj poruka, njihova sintaksa i semantika su prethodno definisane [2]. Sigurnost u komunikaciji i konzistentnost razmene poruka obezbeđena je korišćenjem HTTPS komunikacionog protokola na ISO OSI nivou sesije. Operacije visokog nivoa koje se mogu podržati na osnovu TR-069 protokola su npr. početna konfiguracija uređaja, praćenje statusa uređaja, postavljanje nove programske podrške na korisničkim uređajima, praćenje vrednosti parametara, postavljanje vrednosti nekih parametara itd. Kao važna stavka svakog modela podataka (eng. *Data Model*) je i definisanje, ne samo bitnih podataka, već i njihovih osobina. Svaki parametar može biti samo za čitanje, samo za upisivanje, ili oba. Na ovaj način se poštuje specifičnost korisničkih uređaja.

2.2 Prilagodni sloj poslužilaca za pristup funkcionalnostima

Funkcionalnosti koje konfiguracioni poslužilac treba da obezbedi korisničkim aplikacijama su definisani TR-131 standardom [3], slično kao komunikacija između korisničkih uređaja i poslužilaca opisan u prethodnom delu teksta. Šemu ovakvog sistema prikazuje Slika 3.



Slika 3. Autokonfiguracioni i nadzorni sistem

Severni prilagodni sloj konfiguracionog poslužilaca (eng. *ACS Northbound Interface - ACS NBI*), gde termin sever označava smer ka korisničkim aplikacijama, po standardu mora da obezbedi tri vrste usluge:

1. Mogućnost dobavljanja informacija o nekom povezanom uređaju ili grupe uređaja
2. Mogućnost slanja zahteva ili konfiguracione poruke nekom povezanom uređaju ili grupi uređaja
3. Mogućnost pretplaćivanja na razne događaje prouzrokovanih od strane povezanih uređaja

U standardu nisu definisani protokol i način povezivanja sa poslužilacom, jedino je zahtevano da konekcija budešifrovana. Ovo nameće korišćenje HTTPS (*Hypertext Transfer Protocol Secure*) protokola za povezivanje korisničke aplikacije i konfiguracionog poslužilaca.

2.3 Mogući pristupi realizaciji korisničke aplikacije

Spektar metoda za razvoj korisničkih aplikacija je širok [4]. Analizom mogućnosti izabrano je nekoliko vrsta koje odgovaraju zahtavima korisničkih aplikacija. Tipovi aplikacija koji su uzeti u obzir pri planiranju opisanog rešenja su prikazani u Tabeli 1.

Tip	Prednosti	Mane
Mobilna aplikacija	<ul style="list-style-type: none"> • Podrška za prenosive uređaje • Pogodna za korišćenje van radnog mesta 	<ul style="list-style-type: none"> • Ograničenja prilikom unosa podataka i navigacije • Ograničen prikaz podataka zbog veličine ekrana
Klasična aplikacija	<ul style="list-style-type: none"> • Dobar odziv na korisničku interakciju • Širok spektar grafičkih elemenata 	<ul style="list-style-type: none"> • Zavisna od platforme • Poteškoće prilikom osvežavanja verzije
Web aplikacija	<ul style="list-style-type: none"> • Mogućnost korišćenja nezavisno od lokacije korisnika • Standardizovana grafička sprega, nezavisna od platforme • Lako razmeštanje (eng. <i>Deployment</i>) i osvežavanje aplikacije 	<ul style="list-style-type: none"> • Potrebna internet konekcija

Tabela 1. Poređenje tipova aplikacija

Najčešći slučaj upotrebe projektovane aplikacije je korisnički centar gde je ključno predstaviti status uređaja pomoću složene grafičke korisničke sprege (eng. *User Interface* - UI) na velikom ekranu, a pri tome da se omogući i prenosivost aplikacije. Analizom ovih činjenica i svojstava različitih tipova aplikacija, odlučeno je da rešenje bude projektovano kao Web aplikacija.

2.3.1 Metode za razvoj Web aplikacija

Za razvoj Web aplikacija postoje razni alati i programski jezici. Najrasprostranjeniji su JavaScript i Java programski jezici. JavaScript jezik, zajedno sa jQuery bibliotekom [5] čine dobar spoj za razvoj jednostavnijih Web aplikacija. Oni omogućuju brz razvoj i lako pronalaženje grešaka, ali kod aplikacija sa složenijom strukturom, one postaju nepregledne i teške za održavanje. Pored toga je potrebno obratiti pažnju i na kompatibilnost sa raznim internet pretraživačima, pa su programeri najčešće prinuđeni da kreiraju i da održavaju po jednu verziju aplikacije za svaki pretraživač.

Java programski jezik sa Google-ovim alatima za razvoj Web aplikacija (*Google Web Toolkit-GWT*) [6] uklanja sve mane prethodno navedene metode. GWT je skup alata, javno dostupnog koda, koji omogućava razvijanje i održavanje složenih JavaScript Web aplikacija u Java programskom jeziku. Takođe, on omogućava ponovnu upotrebu koda (eng. *Code reuse*), jednostavnu realizaciju poziva udaljenih metoda (*Remote Procedure Call-RPC*), apstrakciju korisničke sprege, kao i internacionalizaciju i prenosivost aplikacija na različite pretraživače.

Korišćenjem GWT alata programeri mogu razvijati i otklanjati greške u web-aplikacijama u Java programskom jeziku korišćenjem Java razvojnih alata po svom izboru. Kada je aplikacija razvijena, GWT prevodilac prevodi Java aplikaciju u samostalne JavaScript datoteke koje su optimizovane i prilagođene ciljnom pregledaču internet stranica.

Glavne komponente GWT-a su:

1. GWT Java u JavaScript prevodilac – omogućava prevođenje izvornog koda iz Java programskog jezika u JavaScript programski jezik.
2. GWT razvojni režim – omogućava programerima da izvršavaju GWT aplikacije u razvojnom režimu, tj. aplikacija se pokreće kao Java aplikacija u Java virtuelnoj mašini bez prevođenja u JavaScript.
3. Biblioteka za emulaciju ugrađenih Java klasa – JavaScript implementacije često korišćenih klasa iz standardne Java biblioteke (kao što je većina *java.lang* paketa i deo *java.util* paketa).
4. GWT Web UI biblioteka – sadrži skup posebnih sprega i klasa za kreiranje komponenti grafičke korisničke sprege.

Karakteristike GWT-a:

1. Dinamične i ponovo upotrebljive komponente;
2. Jednostavan RPC mehanizam;
3. Upravljanje istorijom pretraživača;
4. Integracija *JUnit* testova;
5. Podrška za otklanjanje grešaka u Java programskom jeziku;
6. Podrška za internacionalizaciju i lokalizaciju;
7. Programeri mogu kreirati svoje aplikacije kao objektno-orijentisane zbog korišćenja Java programskog jezika.

2.4 Mehanizam za udaljeni poziv metoda poslužilaca u GWT aplikaciji

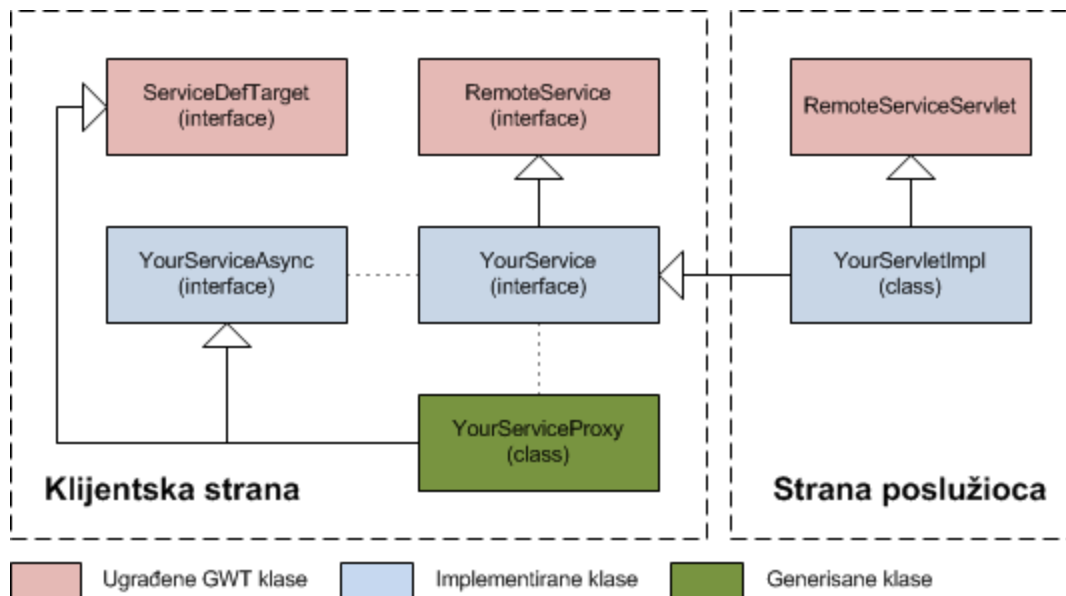
GWT nudi nekoliko načina komunikacije sa poslužilacom putem HTTPS sprege. Najpogodnije je korišćenje ugrađenog GWT RPC mehanizma koji obezbeđuje pozive ka Java servletima i rukovanje detaljima niskog nivoa, kao što je serijalizacija objekata.

Kreiranje RPC-a se sastoji iz sledećih koraka:

1. Definisavanje sprege koja nasleđuje ugrađenu GWT spregu `RemoteService`;

2. Definisane klase koja implementira pozive metoda poslužilaca. Ova klasa treba da nasleđuje ugrađenu GWT klasu RemoteServiceServlet i da implementira spregu koja je kreirana pod 1;
3. Definisane asinhrona spregu koja će se pozivati sa klijentske strane.

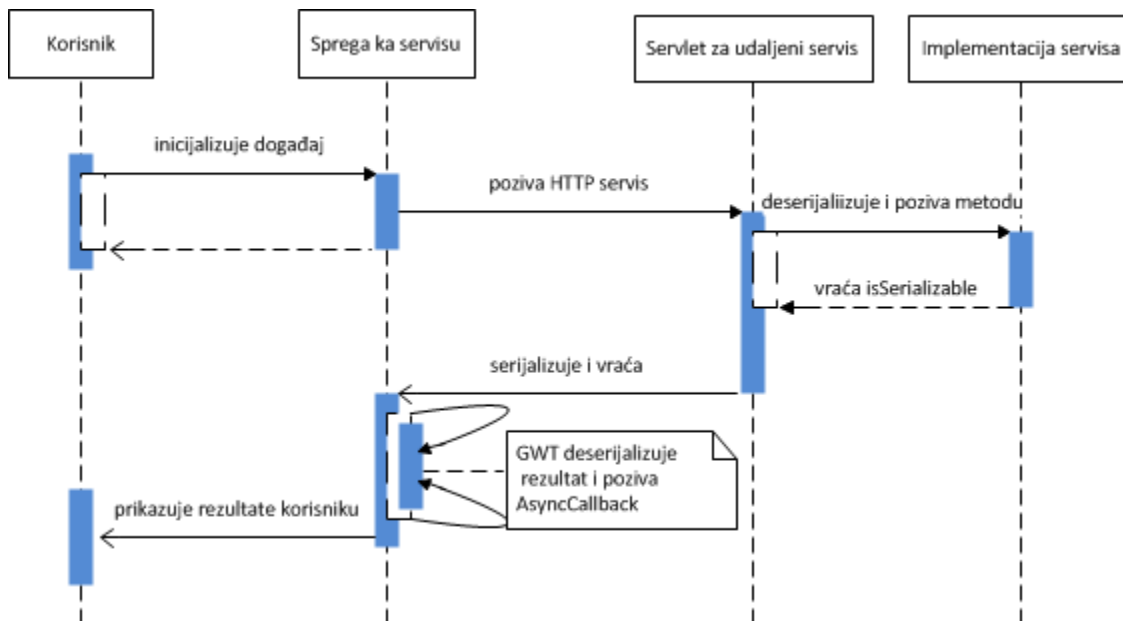
Grafički prikaz realizacije poziva udaljene metode prikazuje Slika 4.



Slika 4. Dijagram realizacije GWT RPC

Da bi se složeni objekti preko mreže poslali, prethodno treba da se serijalizuju. GWT pruža mehanizam serijalizacije putem ugrađene GWT spregu Serializable ili IsSerializable. Da bi se neki objekat serijalizovao, potrebno je implementirati jednu od sprega.

Redosled komunikacije klijentske aplikacije sa poslužilacom prikazuje Slika 5.

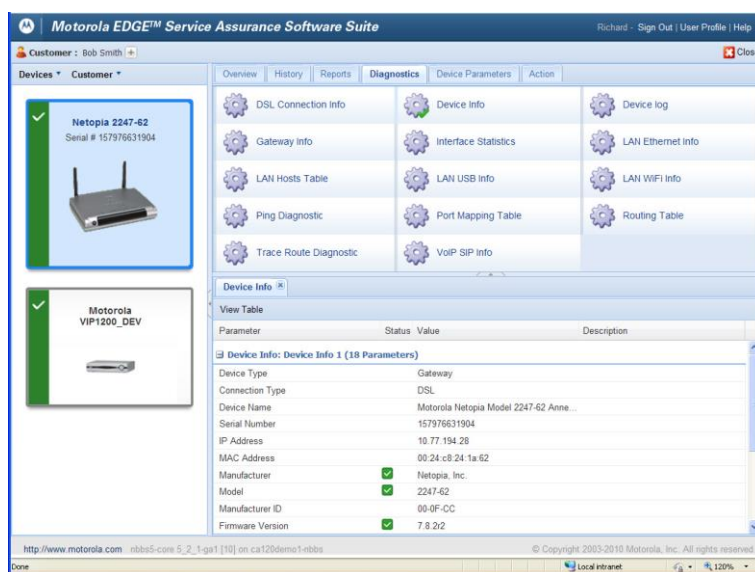


Slika 5. Redosled komunikacije sa poslužilacom

2.5 Postojeća rešenja

Zahtevi provajdera za praćenje i upravljanje mrežom uređaja preko TR-069 protokola su prouzrokovali pojavljivanje raznih autokonfiguracionih poslužioaca u proteklih nekoliko godina. Najpoznatija rešenja za nadgledanje uređaja, kao i ona koja se najviše koriste, su između ostalih Motorola EDGE [7] (Slika 6) i Friendly Technologies TR-069 Device Management Suite [8] (Slika 7). Ova rešenja se ističu dobrim performansama i jednostavnošću korišćenja.

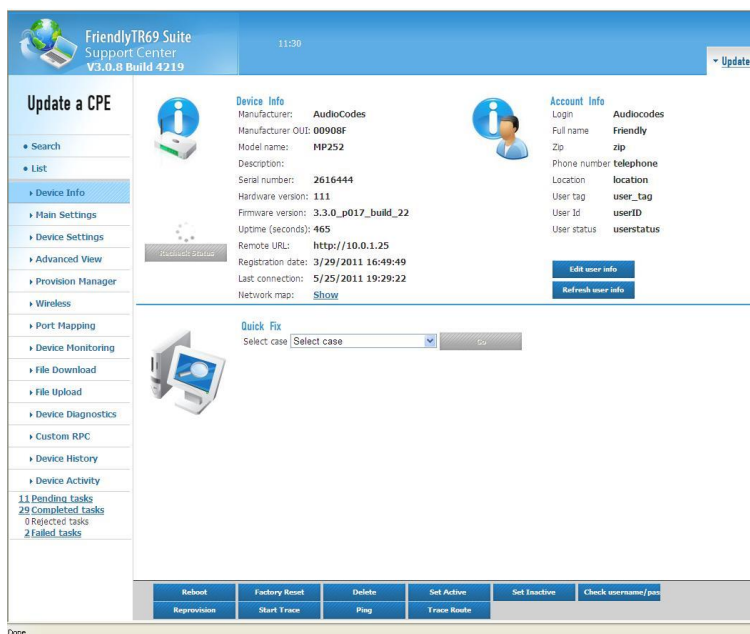
Dobre osobine Motorola EDGE autokonfiguracionog poslužioaca su podrška za veliki broj povezanih uređaja, lak način korišćenja osnovnih funkcionalnosti i moderan dizajn grafičke korisničke sprege. Jedna od mana ovog rešenja je nedostatak podrške za različite načine prikaza podataka i nedostatak mogućih operacija za različite tipove povezanih uređaja. Prikazana je samo generička stranica za svaki tip uređaja, što otežava pristup specifičnim parametrima i funkcionalnostima uređaja. Drugi nedostatak je onemogućen pristup povezanim uređajima od strane krajnjih korisnika. Što znači da krajnji korisnik nema mogućnost nadgledanja svojih uređaja. Takođe nije u stanju da samostalno popravi svoj uređaj, već mora u slučaju greške da kontaktira korisničku podršku.



Slika 6. Korisnička aplikacija Motorola EDGE poslužioaca

Što se tiče pristupa povezanim uređajima, rešenje kompanije Friendly Technologies je bliže krajnjim korisnicima od Motorole. Preko namenskog portala za pristup sopstvenim uređajima, njihovo rešenje omogućuje korisnicima da nadgledaju i podešavaju željene uređaje i da po potrebi rešavaju probleme koji su nastali na povezanim uređajima. Ova mogućnost rasterećuje operatorsku službu korisničke podrške i omogućuje korisnicima brže rešavanje problema. Mana ovog rešenja je, kao i kod Motorole, nedostatak podrške za različite načine prikaza podataka i nedostatak mogućih operacija za različite tipove povezanih uređaja. Umesto

toga se prikazuje generička stranica za svaki tip uređaja, što čini pristup specifičnim parametrima i funkcionalnostima uređaja komplikovanijim.



Slika 7. Korisnička aplikacija poslužioca kompanije Friendly Technologies

U realizovanom rešenju su projektovani mehanizmi za omogućavanje pristupa krajnjim korisnicima i za podržavanje podešavanje izgleda stranica za pregled statusa uređaja. Ovi mehanizmi čine unapređenje u odnosu na postojeća rešenja.

3. Koncept rešenja

3.1 Strukturni zahtevi i slučajevi korišćenja

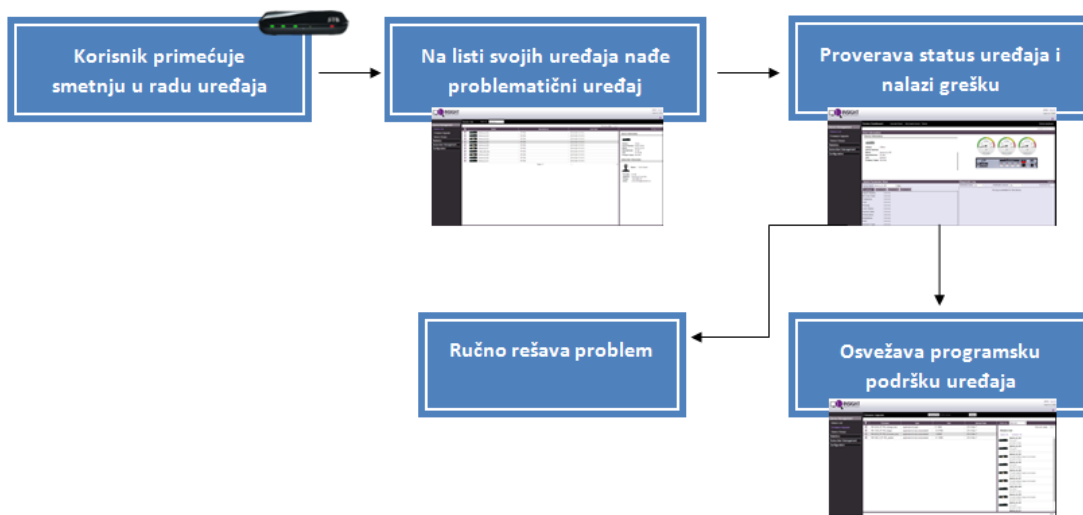
Analizom postojećih rešenja i tehničkog izveštaja TR-131 [3] mogu se utvrditi strukturni i funkcionalni zahtevi za korisničku aplikaciju koja treba da omogući pristup funkcionalnostima poslužilaca za automatsku konfiguraciju TR-069 uređaja. Funkcionalni zahtevi su predstavljeni i u izveštaju Broadband Foruma [9] pod nazivom MR-230, koji opisuje scenarije korišćenja TR-069 protokola, a samim tim i konfiguracionog poslužioca.

Mogu se uočiti sledeći strukturni zahtevi:

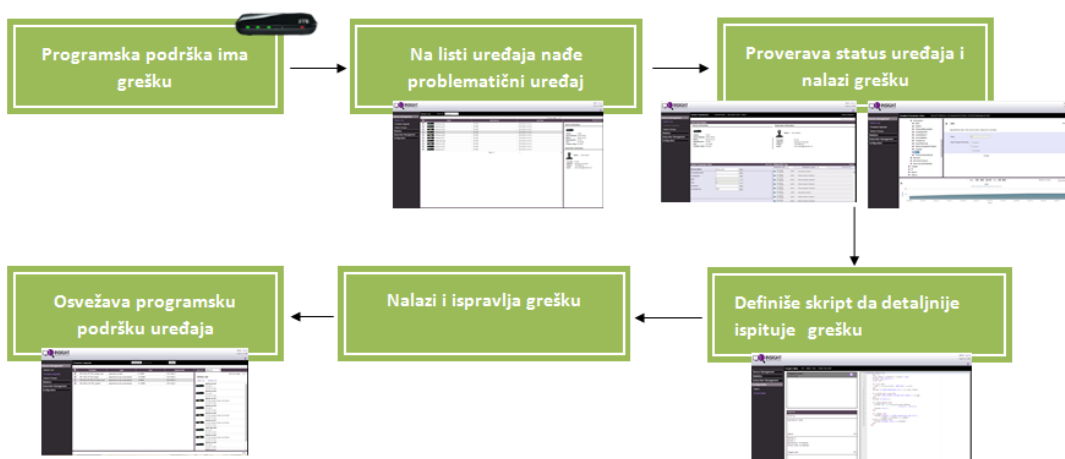
- SZ1. Modularnost** - Aplikacija treba da se sastoji od jasno definisanih i odvojenih modula radi lakše izmene i dodavanja komponenti.
- SZ2. Prenosivost** - Poželjno je da aplikacija radi podjednako dobro na što većem broju internet pretraživača. Ovaj zahtev je zadovoljen korišćenjem GWT alata, koji automatski prilagođava prevedenu aplikaciju ciljnim pretraživačima.
- SZ3. Upotrebljivost** - Aplikacija treba da ima jasno definisan i dosledan izgled, da bude intuitivna i laka za korišćenje, a da pri tome prikazuje i sve bitne informacije o korisničkim uređajima.
- SZ4. Robustnost** - Aplikacija treba da bude robustna na greške, tj. da se ne blokira prilikom greške. Takođe je potrebno da se korisnik u slučaju greške obavesti o uzroku i tipu greške, kao i da dobije predlog za rešavanje iste.

Analizom zahteva uočavaju se i različiti tipovi korisnika i njihovi načini korišćenja aplikacije. Korisnici aplikacije su:

1. Korisnik uređaja–Ovakvih korisnika postoji najviše u sistemu. Svaki korisnik je u mogućnosti da vidi status i da konfigurira svoje uređaje (Slika 8).
2. Inženjer razvoja – Inženjeri razvoja su zaduženi za uklanjanje grešaka iz programske podrške uređaja i unapređenje iste (Slika 9).
3. Viši menadžer– Menadžeri provajdera su zaduženi za odluke u vezi sadržaja i unapređenja kvaliteta pruženih usluga (Slika 10). Za to im je potreban uvid u statistiku sistema.
4. Operater korisničke podrške– Operateri imaju dva zadatka. Prvi je registracija novih korisnika i povezivanje njihovih naloga sa uređajima koji su u mogućnosti da nadgledaju i konfiguriraju pomoću konfiguracionog poslužilaca (Slika 11). Druga uloga je uklanjanje smetnji i grešaka u sistemu i na uređajima koje korisnici krajnjih uređaja nisu u mogućnosti samostalno da reše (Slika 12).



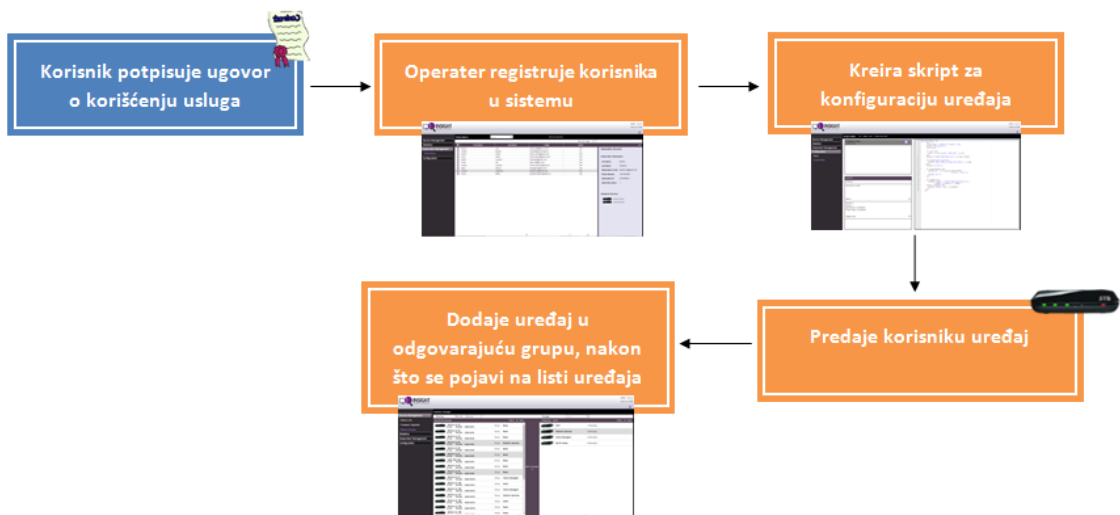
Slika 8. Koraci korisnika pri rešavanju problema



Slika 9. Koraci za nalaženje greške u programskoj podršci uređaja



Slika 10. Koraci za analizu stanja sistema

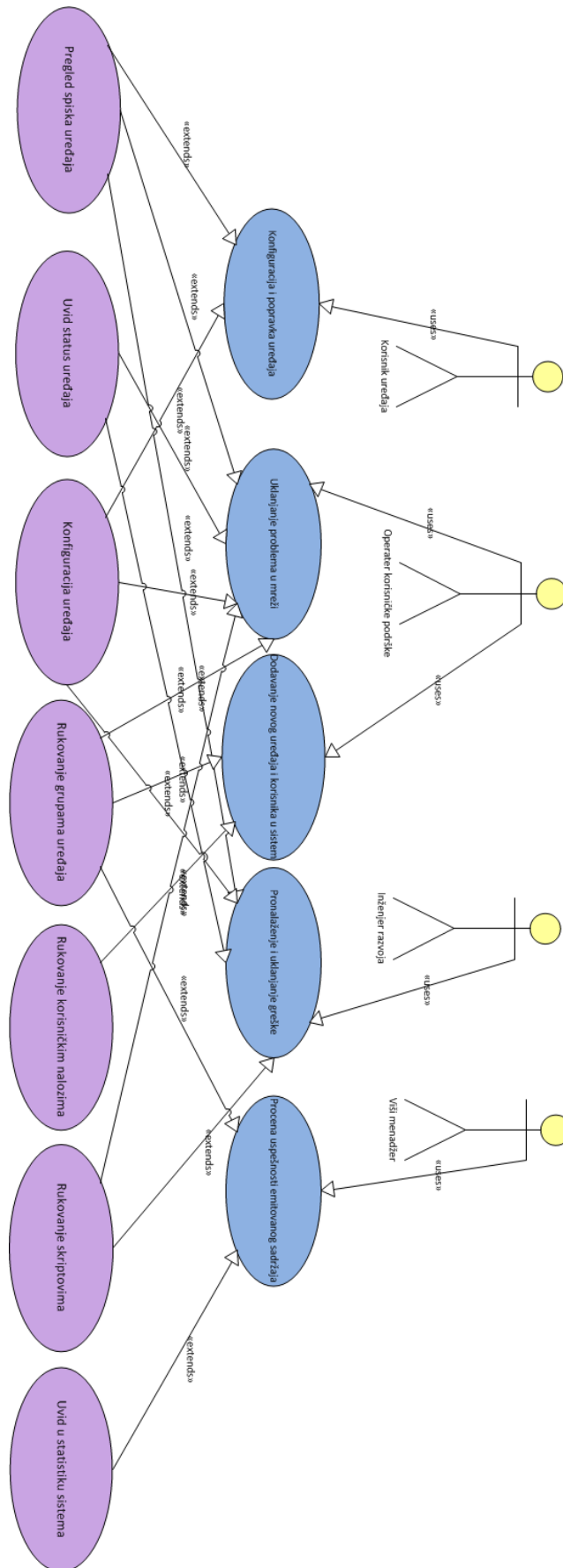


Slika 11. Koraci za registraciju novog korisnika



Slika 12. Koraci rada korisničke podrške pri rešavanju korisničkog problema

Slučajevi korišćenja aplikacije iz aspekta svih korisnika prikazuje Slika 13.



Slika 13. Slučajevi korišćenja aplikacije

Detaljni spisak svih slučajeva korišćenja je sledeći:

- SK1. Pregled spiska uređaja** - Mogućnost listanja povezanih uređaja i pretraživanje ovakve liste.
- SK2. Uvid u status uređaja** - Mogućnost praćenja statusa i vrednosti parametara povezanih uređaja.
 - SK2.1.** Jasan prikaz statusa i aktivnosti uređaja.
 - SK2.2.** Brz prikaz bitnijih parametra uređaja. Takođe je potrebno dati mogućnost da korisnik sam izabere koje parametre želi da se prikažu.
 - SK2.3.** Detaljan prikaz vrednosti svih parametra sa dodatnom istorijom promena tokom određenog vremenskog perioda.
 - SK2.4.** Prikaz istorije događaja (eng. *Log*) za date uređaje.
- SK3. Konfiguracija uređaja**
 - SK3.1.** Mogućnost slanja zahteva za dobavljanje i postavljanje vrednosti parametara.
 - SK3.2.** Mogućnost slanja komande za ažuriranje programske podrške.
 - SK3.3.** Mogućnost ponovnog pokretanja i vraćanje vrednosti parametara datog uređajana fabrička podešavanja.
- SK4. Rukovanje grupama uređaja** - Potrebno je omogućiti organizaciju uređaja u grupe i pretraživanje iste.
- SK5. Rukovanje korisničkim nalozima**
 - SK5.1.** Mogućnost listanja i pretraživanja korisnika.
 - SK5.2.** Mogućnost dodavanja, brisanja i izmene korisnika uređaja.
- SK6. Rukovanje skriptovima** - Podrazumeva dodavanje, brisanje i izmenu skripti koje se izvršavaju prilikom dešavanja određenog događaja.
- SK7. Uvid u statistiku sistema** - Prikaz statistike o korišćenju uređaja i vrednosti njegovih parametara.

3.2 Struktura Web aplikacije

Iz zahteva i slučajeva korišćenja u prethodnom poglavlju (Poglavlje 3.1) direktno proizilazi struktura projektovane aplikacije. Struktura Web aplikacije je sledeća:

1. Spisak uređaja;
 - 1.1. Pregled statusa uređaja;
 - 1.1.1. Informacije o uređaju
 - 1.1.2. Informacije o korisniku uređaja

- 1.1.3. Uvid u parametre uređaja
 - 1.1.3.1. Uvid u odabrane parametre
 - 1.1.3.2. Detaljni pregled parametara uređaja
 - 1.1.3.3. Izmena vrednosti parametra
 - 1.1.3.4. Uvid u istoriju vrednosti parametra
2. Osvežavanje programske podrške uređaja
 - 2.1. Rukovanje paketima programske podrške
 - 2.2. Primena paketa programske podrške na jedan ili više uređaja
3. Rukovanje grupama uređaja
4. Mapa za prikaz distribucije vrednosti parametra
5. Pozicije uređaja u nadgledanom području
6. Statistika sistema
 - 6.1. Gledanost kanala
 - 6.2. Aktivnost uređaja u vremenu
7. Rukovanje nalogima vlasnika uređaja
8. Rukovanje korisničkim nalogima aplikacije
9. Rukovanje skriptovima

Da bi se zadovoljili strukturni zahtevi za modularnost i robustnost, potrebno je podeliti aplikaciju na posebne module koji čine jednu celinu i ne zavise od drugih komponenti. Komunikacija između ovih modula treba takođe da bude standardizovana [10]. Iz ovih razloga je definisan jedan centralni modul koji upravlja radom ostalih modula. Takođe su jasno razdvojene komponente za rukovanje i za prikazivanje podataka.

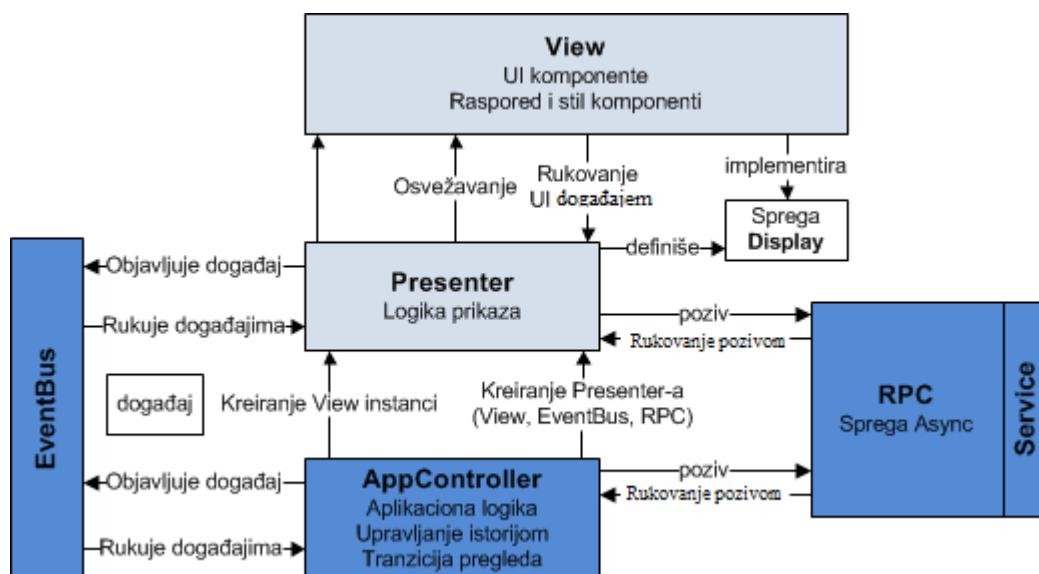
Osnovni elementi Web aplikacije su:

1. **AppController** – Da bi se upravljalo logikom koja nije namenjena za pojedinačne module aplikacije i koja pripada aplikativnom sloju uvodi se komponenta **AppController**. Ova komponenta omogućava upravljanje istorijom i promenom stranica pretraživača. Tranzicija pogleda je direktno povezana sa upravljanjem istorijom. U ovoj klasi se definišu i povezuju sve potklase klase *View* i *Presenter*.
2. **EventBus** – Pri kreiranju složenih aplikacija jedan od osnovnih ciljeva je modularnost, tj. zaokruživanje funkcionalnosti jednog dela aplikacije u celinu, tako da jedan deo aplikacije ne zna kako ostali delovi funkcionišu. Modularnost se često narušava kada moduli trebaju da međusobno komuniciraju. Jedan od rešenja ovog problema je korišćenje GWT mehanizma za prenos događaja (eng. *Event Bus*) mehanizma koji omogućava komunikaciju između modula. **EventBus** predstavlja

magistralu kroz koju se šalju događaji (eng. *Events*). Komunikacija se zasniva na tome da modul koji objavi događaj ne zna kome je on namenjen, već zainteresovani moduli trebaju da se pretplate na događaje. Događaj se sastoji iz dva dela:

- a. Klase koje sadrže opis događaja: pri objavljivanju događaja, objekat klase će se proslediti na magistrali događaja.
 - b. Sprege koje sadrže metode koje implementiraju moduli pretplaćeni na dati događaj.
3. **StatusUpdater** – Ovaj modul omogućuje automatsko ažuriranje prikazanih podataka u modulima aplikacije. StatusUpdater sadrži časovnik čiji se period podešava pomoću parametra pri registrowanju komponente za osvežavanje. Nakon isteka časovnika upoređuju se podaci sa klijentske strane i podaci sa strane poslužioca. Ako je došlo do promene neke vrednosti, poziva se metoda registrovanog *Presenter*-a za osvežavanje prikazanih podataka.
 4. **Display** – Zadatak ovog modula je povezivanje modula *View* i *Presenter*. Povezivanje modula *View* i *Presenter* se sastoji iz dva koraka. *Presenter* definiše Display spregu sa svim potrebnim metodama, a potom modul *View* implementira definisane metode.
 5. **Presenter** – Zadatak ovog modula je povezivanje modela podataka sa grafikom vidljivom na ekranu. U okviru ovog modula vrši se komunikacija sa poslužiocom putem RPC sprege, obrađuju se događaji i upravlja se logikom UI komponenti.
 6. **View** – Ovaj modul predstavlja deo aplikacije koji je vidljiv na ekranu. Komponente modula *View* se kreiraju pomoću GWT alata *UiBinder*. *UiBinder* omogućuje mehanizam povezivanja deklarativnog opisa korisničke sprege sa kodom pisanim u Java jeziku. Prednost korišćenja alata *UiBinder* je jednostavnost projektovanja i izmene korisničke sprege.

Strukturu korisničke aplikacije sa rasporedom osnovnih elemenata i vezama između njih prikazuje Slika 14.



Slika14. Raspored modula u korisničkoj aplikaciji i veze između njih.

4. Realizovano rešenje

Ovo poglavlje opisuje proces razvoja Web aplikacije kao pristupne tačke funkcionalnostima TR-069 konfiguracionog poslužilaca. Postupak razvoja Web aplikacije je dat na reprezentativnom primeru u poglavlju koje opisuje raspored modula iz razloga što se za razvoj svih modula aplikacije koriste slične metode. Umesto toga će biti izložene stranice Web aplikacije u vidu izgleda i opisa funkcionalnosti. Na kraju poglavlja se mogu naći specifična rešenja koja su zahtevala odstupanje od normalnog toka razvoja.

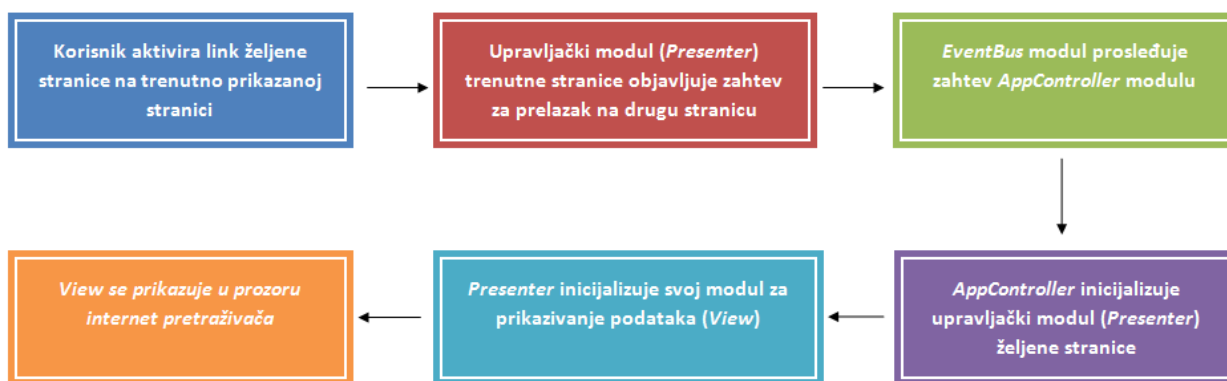
4.1 Raspored modula

Paketi koji sadrže GWT Java klase su sledeći:

1. **com.rtrk.insight.acs.gwt.client.engine** - Sadrži klase za kontrolisanje cele aplikacije, kao i bazne klase od kojih se prave *View - Presenter* parovi.
2. **com.rtrk.insight.acs.gwt.client.presenter** - Sadrži klase koje upravljaju podacima i *View* klasama u aplikaciji.
3. **com.rtrk.insight.acs.gwt.client.view** - Sadrži klase koje su zadužene za raspored elemenata i prikazivanje podataka.
4. **com.rtrk.insight.acs.gwt.client.view.widget** - Sadrži kreirane namenske elemente za operacije koje se ponavljaju u celoj aplikaciji.
5. **com.rtrk.insight.acs.gwt.client.shared** - Ovde se nalaze klase koje se šalju i primaju kroz GWT RPC mehanizam.
6. **com.rtrk.insight.acs.gwt.client.async** - Klase koje služe kao posrednik između Web aplikacije i konfiguracionog poslužilaca su smeštene u ovaj paket. One opisuju metode poslužilaca u vidu parametra i povratne vrednosti.

Dijagram klasa sa reprezentativnim skupom *View - Presenter* parova prikazuje Slika 10.

Na Slici 15 se vidi da je centralna tačka aplikacije klasa *AppController*. Ova komponenta instancira i povezuje ostale elemente Web aplikacije. *AppController* je takođe zadužen za navigaciju kroz aplikaciju. *AppController* proširuje baznu klasu *Presenter* da bi se obezbedili mehanizmi za osvežavanje i za pretplaćivanje na događaje u sistemu. Osvežavanje ove komponente se radi zbog provere aktuelnosti sesije korisnika. Nakon isteka korisničke sesije, potrebno je ponovno prijavljivanje u aplikaciju. Pretplaćivanje na događaje je takođe važan mehanizam pomoću koga se dobija informacija o zahtevima za promenu stranice. U slučaju ovakvog događaja, *AppController* inicira željenu stranicu. Postupak učitavanja nove stranice je prikazan na Slici 16.



Slika 16. Postupak učitavanja nove stranice

Svaki *Presenter* u sistemu (na Slici 15: *ModifyDevicePresenter*, *GroupManagerPresenter*, *DeviceManagerPresenter* itd.) nasleđuje baznu klasu *Presenter* iz istih razloga kao i u slučaju *AppController*-a. U okviru *Presenter* klase se definiše Java Display interfejs, koji služi za komunikaciju sa određenim *View* klasama (na Slici 15: *ModifyDeviceView*, *GroupManagerView*, *DeviceManagerView*, itd.).

Klase paketa *View*, za razliku od *Presenter*-a, se ne instanciraju unapred u okviru *AppController* modula, već samo one koje nemaju pretka u hierarhiji aplikacije, već ih *AppController* prikazuje na zahtev. Takve su npr. klase *ModifyDeviceView* i *GroupManagementView* prikazane na Slici 10. Ovi elementi predstavljaju stranice Web aplikacije. Gradivni elementi stranica, npr. *DeviceManagerView* potklasa klase *GroupManagementView*, se instanciraju u okviru klase sadržioca pomoću GWT alata *UiBinder*. Ovaj alat omogućuje definisanje sadržaja stranice pomoću deklarativnog opisa u XML formatu i povezivanje sa Java kodom. Ovo se postiže definisanjem standardnog Java *Interface*-a unutar klase čiji izgled želimo definisati pomoću *UiBinder*-a. Primer *UiBinder* deklaracije prikazuje Slika 17.

```
<gwt:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'  
  xmlns:gwt='urn:import:com.google.gwt.user.client.ui'>  
  <gwt:VerticalPanel>  
    <gwt:HorizontalPanel>  
      <gwt:Label text="Login" />  
      <gwt:TextBox ui:field="loginBox" />  
      <gwt:Label ui:field="completionLabel1" />  
    </gwt:HorizontalPanel>  
    <gwt:HorizontalPanel>  
      <gwt:Label text="Password" />  
      <gwt:TextBox ui:field="passwordBox" />  
      <gwt:Label ui:field="completionLabel2" />  
    </gwt:HorizontalPanel>  
    <gwt:Button ui:field="buttonSubmit" text="Submit" />  
  </gwt:VerticalPanel>  
</gwt:UiBinder>
```

Slika 17. Primer UiBinder deklarativnog opisa

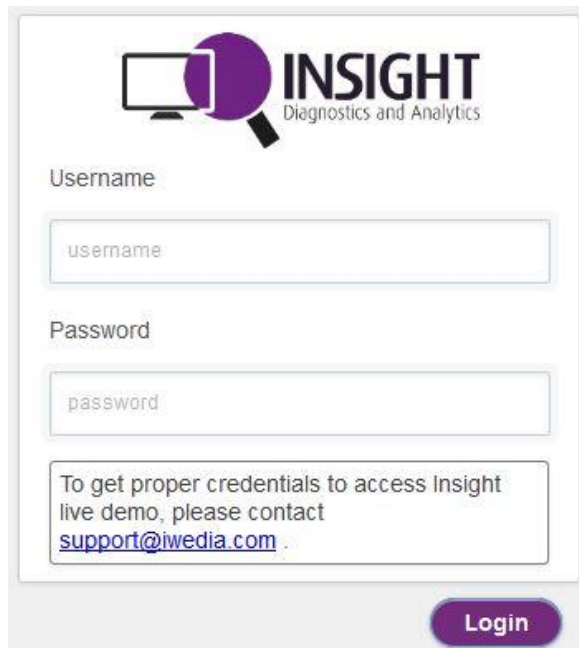
Pomoću *UiBinder* alata je takođe moguće definisati nove grafičke elemente koji se mogu koristiti kao ugrađeni GWT gradivni blokovi (npr. labela, dugme itd.). Primer ove funkcionalnosti na Slici 15 je klasa *PagingControl*.

Aplikacija sadrži po jednu instancu klasa *AppEventBus* i *StatusUpdater* koji, pored korisničkih interakcija, takođe upravljaju izvršavanjem Web aplikacije. *AppEventBus* je zadužen za rukovanje događajima u aplikaciji. *StatusUpdater* služi za automatsko osvežavanje registrovanih modula aplikacije, a detaljniji opis se nalazi u daljem tekstu.

4.2 Izgled i funkcionalnosti realizovane aplikacije

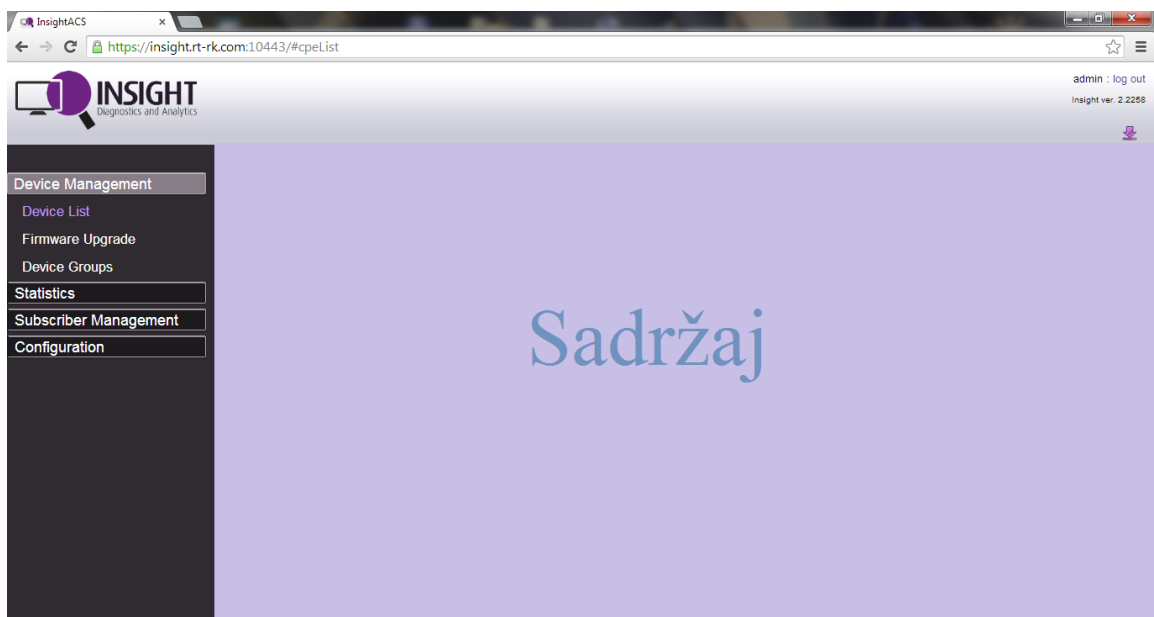
4.2.1 Globalni izgled

Kod pokretanja Web aplikacije iz pretraživača najpre se prikazuje stranica za prijavljivanje u sistem. Ovaj mehanizam je direktno vezan za aplikacioni server koji pokreće Web aplikaciju. Jedino je izgled HTML stranice za prijavljivanje prilagođen izgledu Web aplikacije. Stranica za prijavljivanje je prikazana na Slici 18.



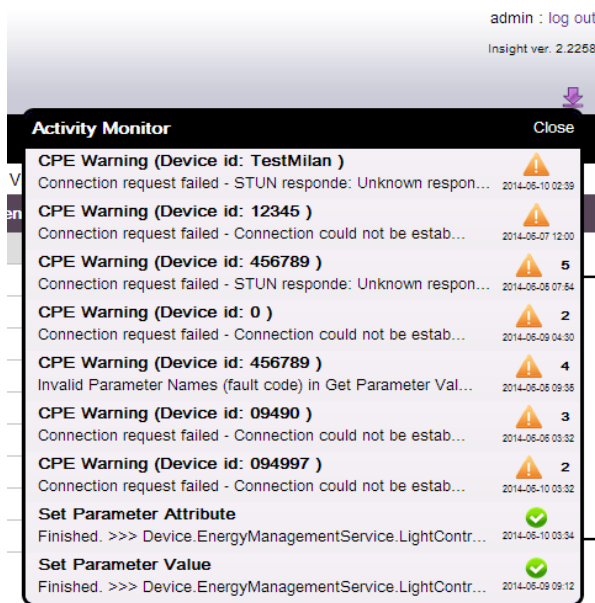
Slika 18. Stranica za prijavljivanje

Posle uspešnog prijavljivanja korisnik dobija pristup Web aplikaciji za automatsku konfiguraciju korisničkih uređaja. Opšti izgled aplikacije je prikazan na Slici 19.



Slika 19. Opšti izgled aplikacije

Aplikacija je koncipirana tako da su meni sa strane i poglavlje na vrhu fiksirani tj. ne učitavaju se prilikom svake izmene stranice. Meni sadrži reference za pristup raznim funkcionalnostima aplikacije. U zaglavlju se nalazi informacija o trenutno prijavljenom korisniku kao i opcija za odjavljivanje. Takođe se ovde nalazi dugme za prikazivanje liste aktivnosti koje se odvijaju na poslužiocu. Prozor aktivnosti se može videti na Slici 20.

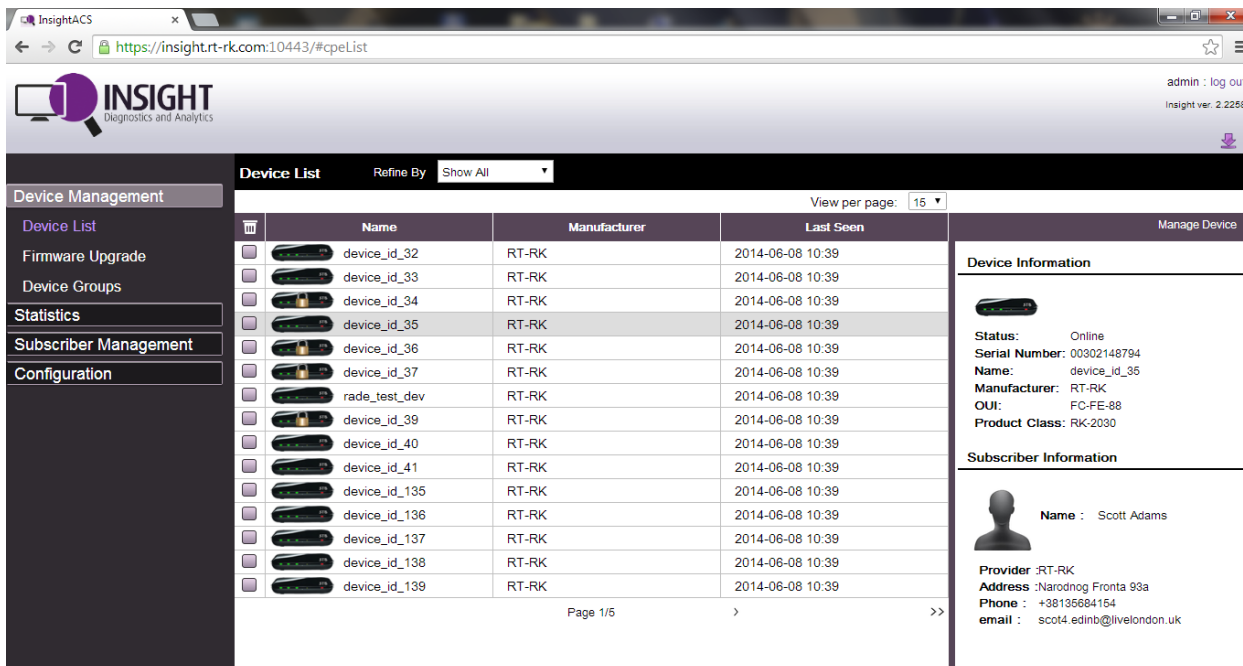


Slika 20. Prozor aktivnosti

Prozor aktivnosti ne može u realnom vremenu da prikazuje dešavanja na poslužiocu zbog nedostatka GWT biblioteke da se pretplati na udaljene događaje na serveru. Zbog toga se notifikacije skupljaju na poslužiocu a Web aplikacija ih periodično zatraži. Perioda očitavanja se menja u zavisnosti od stanja aplikacije.

4.2.2 Spisak uređaja

Stranica za pregledanje i pretraživanje spiska uređaja je prikazana na Slici 21.



Slika 21. Spisak uređaja

Stranica omogućuje listanje i pretraživanje povezanih uređaja. Pretraživanje se vrši izborom kriterijuma za filtriranje i upisivanjem ključne reči u polje pretraživanja. Posle toga aplikacija formira poruku za upit u bazi podataka poslužilaca i nakon dobijenog odgovora prikazuje rezultat pretrage. Takođe je moguće izabrati broj uređaja po stranici što omogućuje prilagođenje aplikacije raznim veličinama ekrana. Kontrola stranice na dnu liste se prilagodi prilikom svake promene kriterijuma pretrage ili izborom drugog broja uređaja po stranici.

Pored liste uređaja su prisutne još dve komponente koje prikazuju osnovne podatke za identifikaciju trenutno izabranog uređaja i njegovog vlasnika. Vezu između ovih komponenti liste uređaja ostvaruje *AppEventBus* modul.

Aktiviranjem hiperveze (eng. *Hyperlink*) u zaglavlju stranice se dolazi do stranice za pregled statusa izabranog uređaja.

4.2.3 Pregled statusa uređaja

Osnovna stranica za pregled statusa uređaja se može videti na Slici 22.

The screenshot displays the 'INSIGHT Diagnostics and Analytics' interface. The main content area is titled 'Device Dashboard' and includes a sidebar with navigation options like 'Device Management', 'Device List', 'Firmware Upgrade', 'Device Groups', 'Statistics', 'Subscriber Management', and 'Configuration'. The dashboard is divided into several sections:

- Brief Information:** Contains 'Device Information' (Status: Online, Serial Number: 00302148794, Name: device_id_35, Manufacturer: RT-RK, OUI: FC-FE-88, Product Class: RK-2030) and 'Subscriber Information' (Name: Scott Adams, Provider: RT-RK, Address: Narodnog Fronta 93a, Phone: +38155684154, Email: scott4.edinb@livelondon.uk).
- Quick Parameter View:** A table of device parameters for 'device_id_35' with input fields and 'Set' buttons. Parameters include Channel/Bandwidth (0), Constellation (0), BER (77), SNR (28), Frequency (0), and CurrentService (TEST).
- Diagnostic Log:** A table showing notification events. The log includes columns for time, notification level, and source. Entries include 'CORE ope_check_and_reboot' and 'COMM Performing session transaction!'.

Slika 22. Osnovnastranica za pregled statusa uređaja

I na ovoj stranici se mogu videti moduli za prikazivanje informacija o uređaju i vlasnika uređaja, što omogućuje modularnost aplikacije iz zahteva SZ1. Ispod panela za informaciju su dva modula za prikaz statusa uređaja.

Desno se nalazi komponenta za prikaz dnevničkih datoteka (eng. *logfile*) koji sadrži pregled dešavanja u programskoj podršci uređaja. Zapisana dešavanja je moguće filtrirati po modulima odakle su zapisani ili po vrsti događaja. Takođe je moguće sa opcijama u zaglavlju modula zatražiti noviju datoteku od uređaja ili sačuvati trenutnu u obliku tekstualne datoteke.

U levom panelu je prikazan modul za brzi prikaz vrednosti parametara uređaja. Ovde imamo mogućnost praćenja vrednosti izabranog skupa parametara. Ovaj skup parametara je moguće izmeniti pritiskom na opciju u zaglavlju modula, čime modul ulazi u stanje za izmenu prikazanih parametara. Ovo stanje je prikazano na Slici 23.

Quick Parameter View		Finish Editing	
Device Name	<input type="text" value="device_id_32"/>	Set	
ChannelBandwidth	<input type="text" value="0"/>	Set	Remove
Constellation	<input type="text" value="0"/>	Set	Remove
BER	<input type="text" value="145"/>	Set	Remove
SNR	<input type="text" value="34"/>	Set	Remove
Frequency	<input type="text" value="0"/>	Set	Remove
CurrentService	<input type="text" value="RTV 1"/>	Set	Remove
Add Parameter			

Slika 23. Stanje za izmenu prikazanih parametara

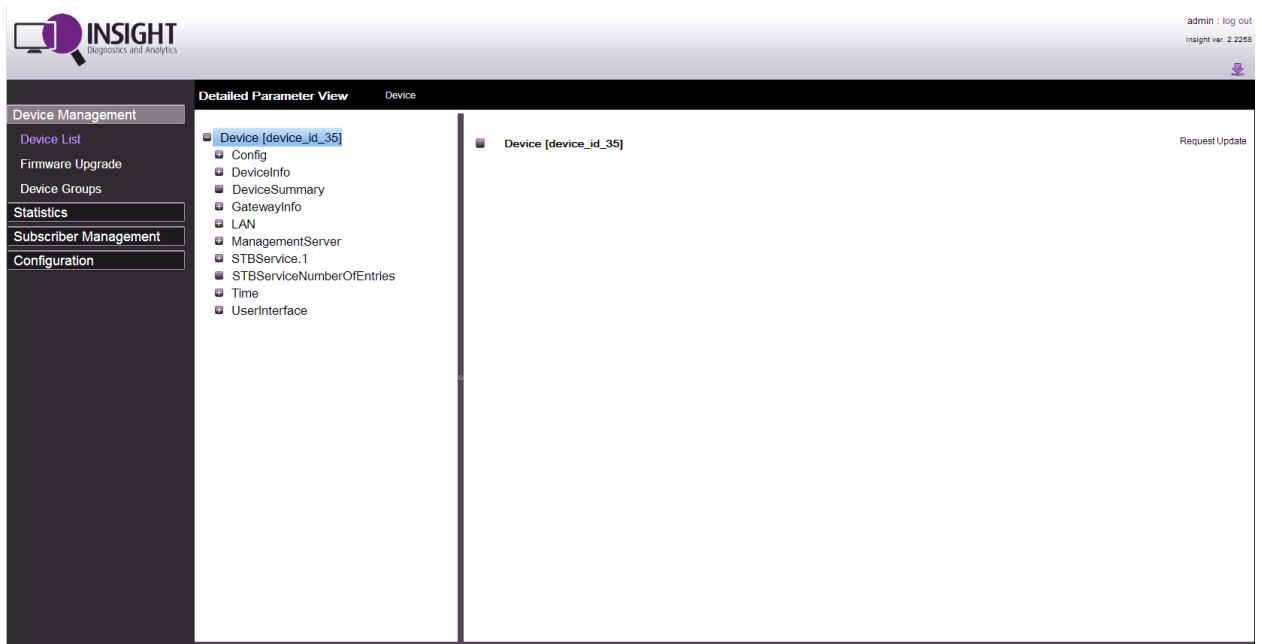
Ovde je moguće obrisati parametre sa spiska ili dodavati nove pomoću dijaloga prikazanog na Slici 24.

Slika 24. Dijalog za dodavanje parametra

Korisnik ima i mogućnost postavljanja vrednosti parametara ukoliko je to moguće za dati parametar. U slučaju postavljanja vrednosti prikazuje se indikator da je zahtev poslat uređaju. Indikator nestaje kad se željena vrednost postavi na uređaju.

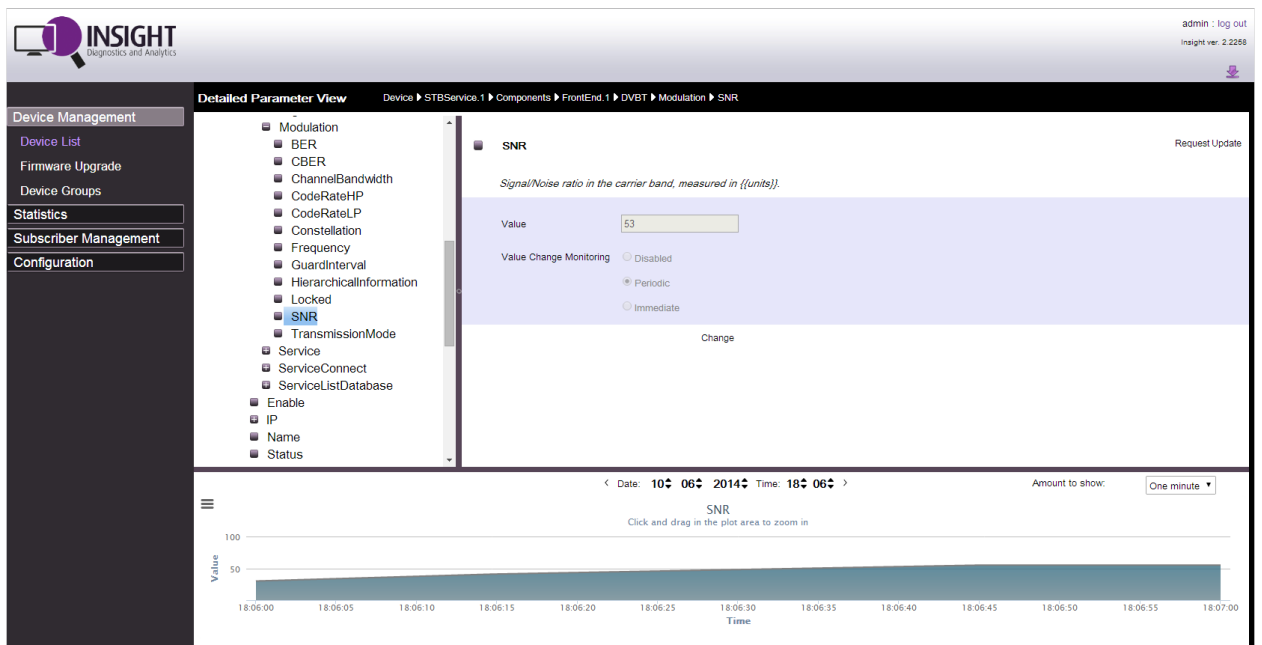
4.2.4 Detaljni pregled parametara

Stranica za detaljni pregled parametara je prikazana na Slici 25.



Slika 25. Stranica za detaljni pregled parametara

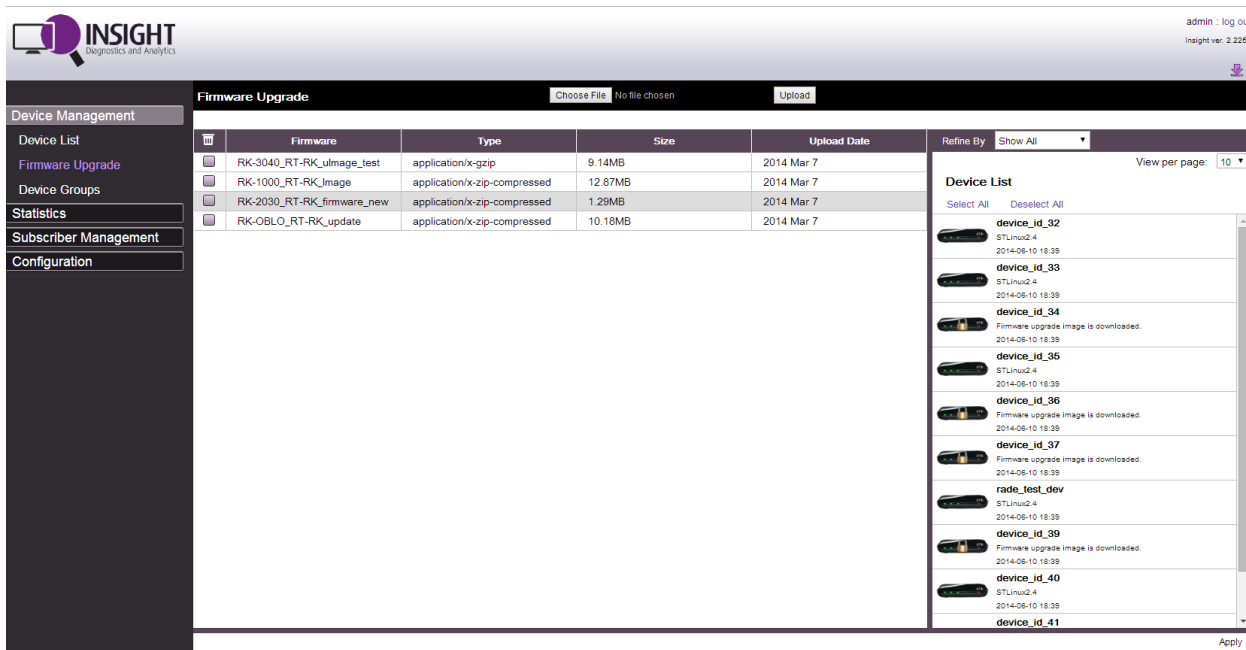
Modul ima iste funkcionalnosti kao prethodno opisani modul za brz prikaz parametara tj. očitavanje i postavljanje vrednosti parametra, sa razlikom da ovde ne postoji spisak izabranih parametara, već se formira stablo svih parametara uređaja. Ovo je moguće korišćenjem GWT komponente *Tree*. Izborom parametra za koji poslužilac čuva istoriju promene vrednosti, u donjem panelu se pojavi modul za prikazivanje ovih promena (Slika 26).



Slika 26. Izabrani parametar sa prikazanim promenama u vremenu

4.2.5 Stranica za osvežavanje programske podrške uređaja

Slika 27 prikazuje stranicu koja služi za osvežavanje programske podrške uređaja.



The screenshot shows the 'Firmware Upgrade' section of the INSIGHT interface. On the left, there is a navigation menu with options like 'Device Management', 'Device List', 'Firmware Upgrade', 'Device Groups', 'Statistics', 'Subscriber Management', and 'Configuration'. The main area is titled 'Firmware Upgrade' and contains a table with the following data:

Firmware	Type	Size	Upload Date
RK-3040_RT-RK_ulmage_test	application/x-gzip	9.14MB	2014 Mar 7
RK-1000_RT-RK_image	application/x-zip-compressed	12.87MB	2014 Mar 7
RK-2030_RT-RK_firmware_new	application/x-zip-compressed	1.29MB	2014 Mar 7
RK-OBLO_RT-RK_update	application/x-zip-compressed	10.18MB	2014 Mar 7

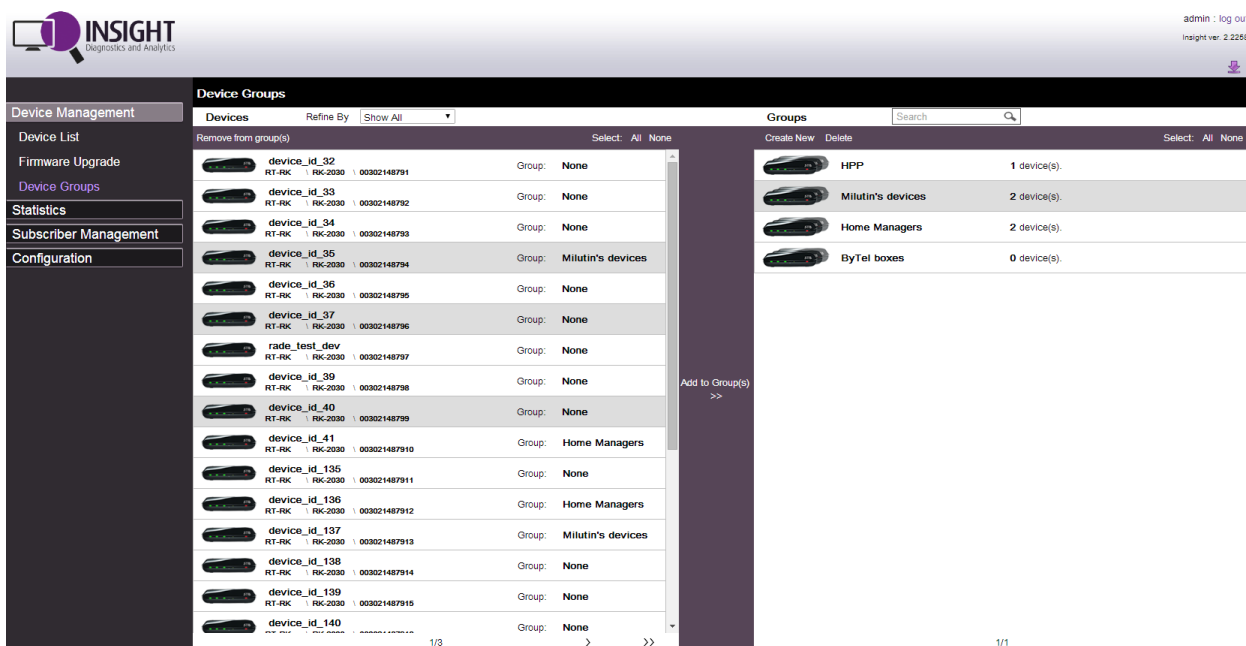
On the right side, there is a 'Device List' section with a 'Refine By' dropdown and a 'Show All' button. The device list includes items like 'device_id_32', 'device_id_33', 'device_id_34', 'device_id_35', 'device_id_36', 'device_id_37', 'rade_test_dev', 'device_id_39', 'device_id_40', and 'device_id_41'. Each item shows its ID, model (STLinux2.4), and a timestamp (2014-06-10 18:39). Some items also indicate 'Firmware upgrade image is downloaded'.

Slika 27. Stranica za osvežavanje programske podrške uređaja

Stranica pruža mogućnost dodavanja i brisanja paketa programske podrške. Spisak uređaja sa strane ima iste funkcionalnosti kao na stranici za pregled spiska uređaja uz manje izmene izgleda. Lista se dodatno filtrira izborom nekog paketa tj. prikazuju se samo uređaji na koje je izabrani paket primenljiv.

4.2.6 Rukovanje grupama uređaja

Stranica koja služi za rukovanje grupama uređaja je prikazana na Slici 28.



The screenshot shows the 'Device Groups' section of the INSIGHT interface. On the left, there is a navigation menu with options like 'Device Management', 'Device List', 'Firmware Upgrade', 'Device Groups', 'Statistics', 'Subscriber Management', and 'Configuration'. The main area is titled 'Device Groups' and contains a table with the following data:

Devices	Refine By	Show All	Groups
device_id_32 RT-RK RK-2030 00302148791			Group: None
device_id_33 RT-RK RK-2030 00302148792			Group: None
device_id_34 RT-RK RK-2030 00302148793			Group: None
device_id_35 RT-RK RK-2030 00302148794			Group: Milutin's devices
device_id_36 RT-RK RK-2030 00302148795			Group: None
device_id_37 RT-RK RK-2030 00302148796			Group: None
rade_test_dev RT-RK RK-2030 00302148797			Group: None
device_id_39 RT-RK RK-2030 00302148798			Group: None
device_id_40 RT-RK RK-2030 00302148799			Group: None
device_id_41 RT-RK RK-2030 003021487910			Group: Home Managers
device_id_135 RT-RK RK-2030 003021487911			Group: None
device_id_136 RT-RK RK-2030 003021487912			Group: Home Managers
device_id_137 RT-RK RK-2030 003021487913			Group: Milutin's devices
device_id_138 RT-RK RK-2030 003021487914			Group: None
device_id_139 RT-RK RK-2030 003021487915			Group: None
device_id_140 RT-RK RK-2030 003021487916			Group: None

On the right side, there is a 'Groups' section with a 'Search' input and a 'Create New' button. The group list includes items like 'HPP', 'Milutin's devices', 'Home Managers', and 'ByTel boxes'. Each item shows its name and the number of devices in the group (e.g., '1 device(s)', '2 device(s)', '2 device(s)', '0 device(s)').

Slika 28. Stranica za rukovanje grupama uređaja

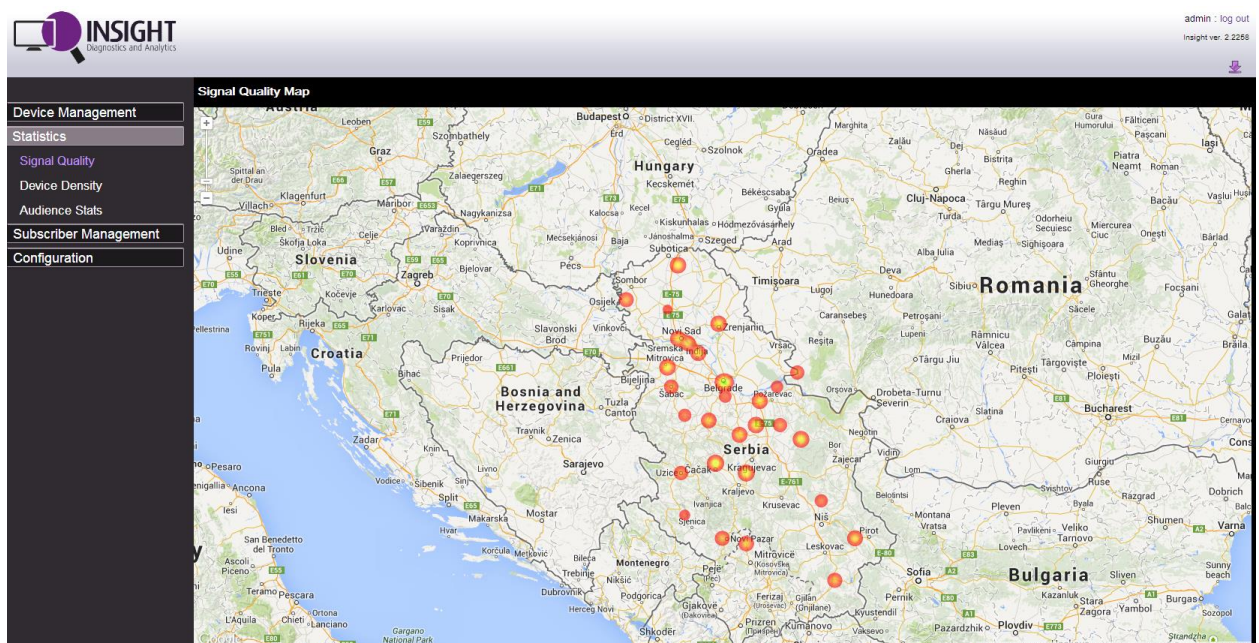
Ovaj modul služi za organizaciju uređaja u grupe radi lakšeg nadgledanja i konfigurisanja. Na primer moguće je grupisati uređaje po lokacijama da bi se konfigurisali po potrebama području kom se nalaze.

Modul u levom panelu služi za listanje uređaja kao što je prikazano na nekoliko prethodnih stranica. Na ovoj stranici se dodatno navodi i informacija o grupama u kojima se uređaj nalazi. Poglavlje liste omogućuje pretraživanje, kao i uklanjanje uređaja iz grupa.

U desnom panelu se prikazuje spisak grupa sa dodatnom informacijom o tome koliko uređaja se nalazi u datoj grupi. Poglavlje ovog modula sadrži opcije za kreiranje, brisanje i pretraživanje grupa.

4.2.7 Mapaza prikaz distribucije vrednosti parametra

U okviru statističkih paketa je realizovan prikaz distribucije vrednosti parametra u odnosu na geografsku lokaciju uređaja. Stranica za ovu svrhu se može videti na Slici 29.

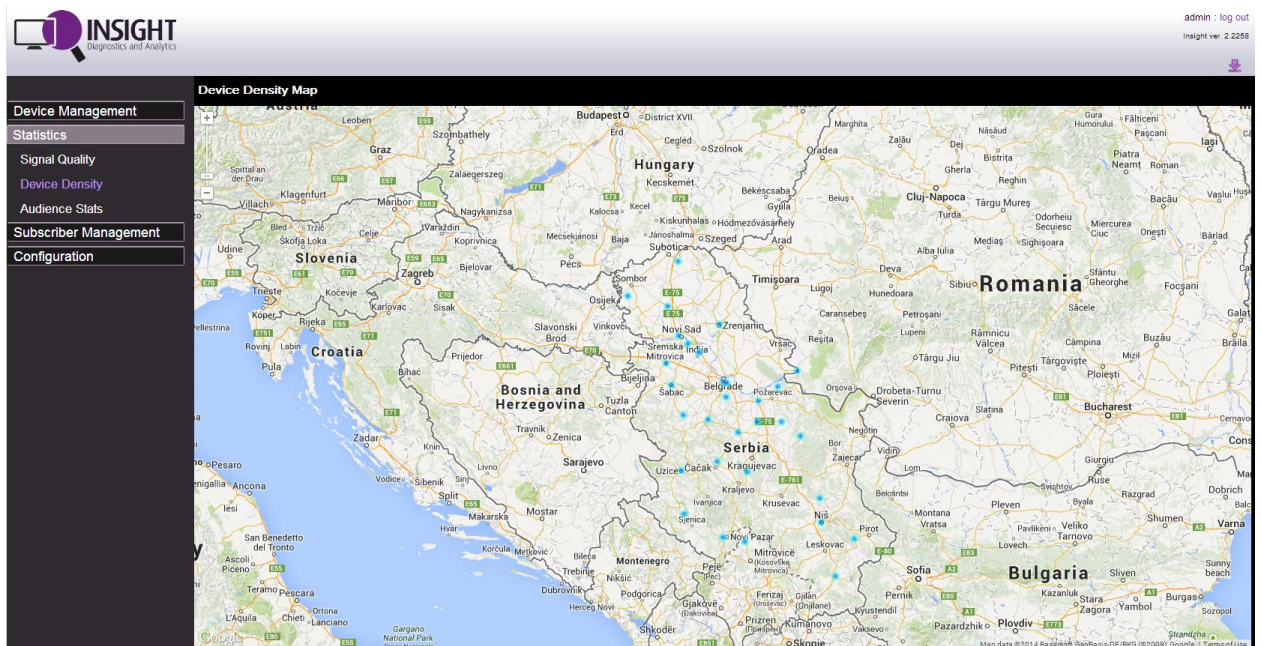


Slika 29. Mapa sa kvalitetom DVB-T2 signala

Aplikacija preuzima podatke o signalu od autokonfiguracionog poslužioaca i prikazuje ih na mapi. Za ovu svrhu je korišćena *GoogleMaps* biblioteka za GWT aplikacije [11] koja već ima ugrađenu podršku za pomeranje i skaliranje mape.

4.2.8 Pozicije uređaja u nadgledanom području

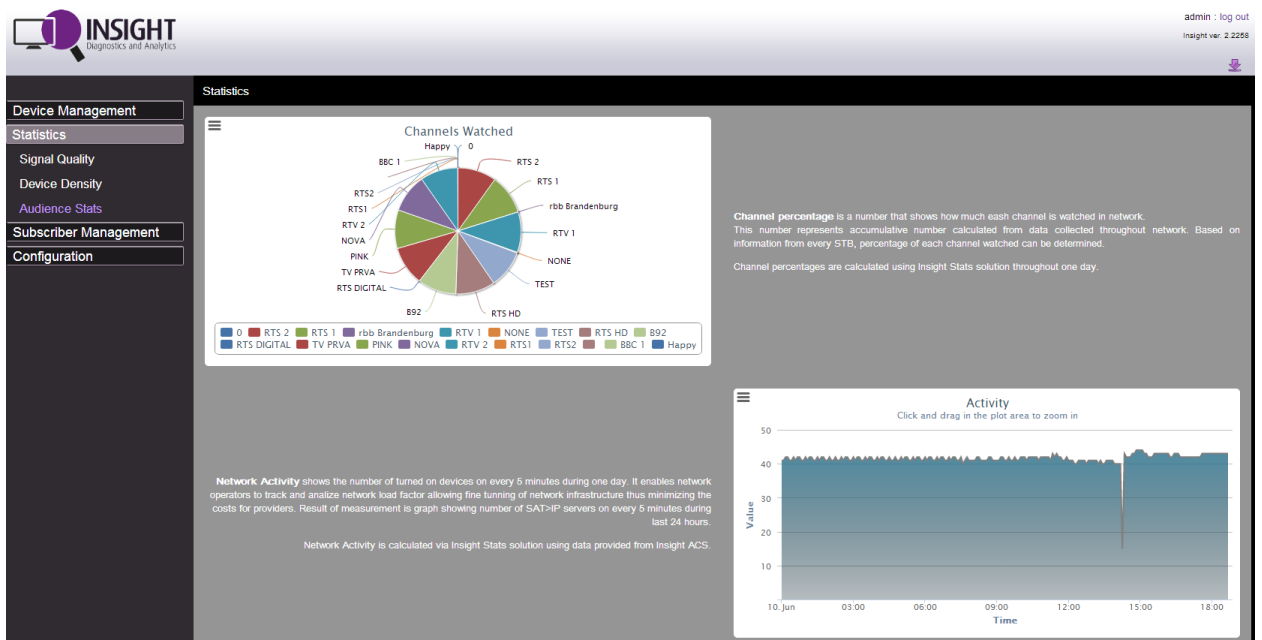
Ova stranica je slična prethodnoj za praćenje kvaliteta signala, samo što umesto toga prikazuje lokacije nadgledanih uređaja. Na osnovu ovih podataka se dobija Slika o gustini uređaja u različitim regionima. Izgled ove mape je prikazan na Slici 30.



Slika 30. Pozicije uređaja u nadgledanom području

4.2.9 Statistika sistema

Stranica za uvid u statistiku sistema je prikazana na Slici 31.



Slika 31. Stranica za uvid u statistiku sistema

Ova stranica prikazuje preuzetu statistiku od poslužioaca u vidu gledanosti kanala dobijeno od STB uređaja (gornji grafikon na Slici 31) i broj aktivnih uređaja u sistemu. Grafikoni su generisani pomoću biblioteke *HighCharts* [12].

4.2.10 Stranica za rukovanje nalozima vlasnika uređaja

Izgled stranice se može videti na Slici 32.

The screenshot displays the 'Subscribers' management page. At the top left is the 'INSIGHT' logo. The main area features a table with the following data:

	First Name	Last Name	E-Mail	Active
<input type="checkbox"/>	Benny	Lee	benny_lee@pesi.com.tw	Yes
<input type="checkbox"/>	Vedran	Bubalo	vedran@hprodukcija.hr	No
<input type="checkbox"/>	Danny	Smith	danny.smith@gmail.com	Yes
<input type="checkbox"/>	Colin	Harris	collin.harris4@yahoo.com	Yes
<input type="checkbox"/>	Martin	Johnson	martinj2@gmail.com	Yes
<input type="checkbox"/>	Frank	Hill	frank.hill@live.com	Yes
<input type="checkbox"/>	Antoan	Jackson	antoan.jackson@bytel.uk	Yes
<input type="checkbox"/>	Mark	Garcia	markgarcia@live.com	Yes
<input type="checkbox"/>	Antony	Anderson	aanthony@gmail.com	Yes
<input type="checkbox"/>	Diana	Miller	dianabirmingham@daily.uk	Yes

On the right side, the 'Subscriber Account' panel for 'Antony Anderson' is visible, showing details like phone number (+3812424562), subscriber ID (21254588241), and assigned devices (two devices with ID 003021487922).

Slika 32. Stranica za rukovanje nalozima vlasnika uređaja

Ovde je omogućen pregled, kreiranje i brisanje korisničkih naloga za vlasnike uređaja. Spisak korisnika ima istu ulogu i funkcionalnosti za korisnike kao lista uređaja za povezane uređaje. Desni panel stranice služi za detaljni prikaz podataka o izabranom korisniku. Takođe je moguće izmeniti podatke o korisnicima. U tom slučaju se prelazi u režim izmene podataka slično modulu za brzi prikaz parametara. Režim za izmenu podataka korisnika je prikazan na Slici 33.

The 'Subscriber Account' edit form contains the following fields and options:

- Subscriber Information:**
 - First Name:
 - Last Name:
 - Subscribers e-mail:
 - Phone Number:
 - Subscribers ID:
 - Subscriber Active:
- Assigned Devices:**
 - Device 1: Remove
 - Device 2: Remove
 - Add:

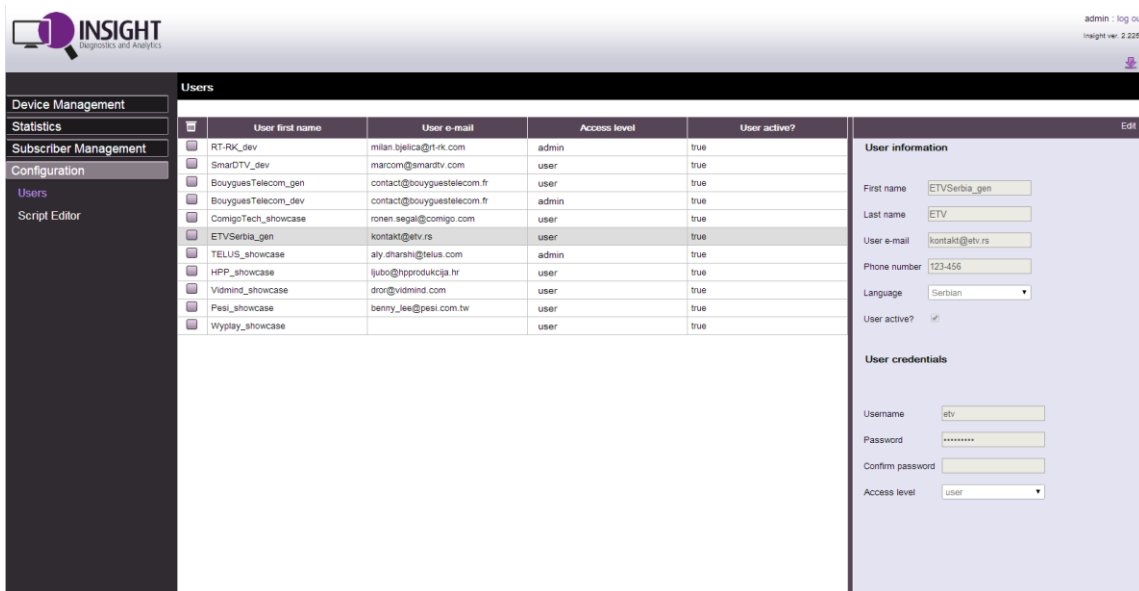
Buttons: Apply, Cancel

Slika 33. Stanje za izmenu podataka korisnika

Pored podataka o korisniku uređaja moguće je dodeliti uređaje upisom serijskog broja uređaja u za to predviđeno polje.

4.2.11 Stranica za rukovanje korisničkim nalozima aplikacije

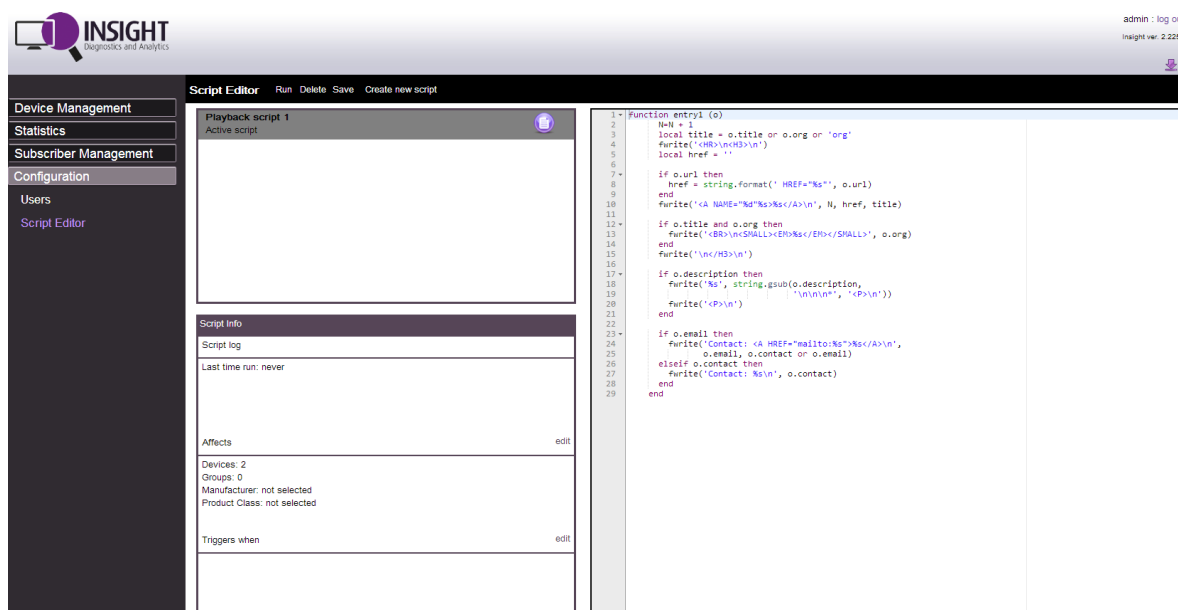
Stranica za rukovanje korisničkim nalozima aplikacije je veoma slična prethodno opisanoj stranici sa razlikom da se ovde rukuje sa korisničkim nalozima za pristup aplikaciji. Izgled ove komponente je prikazan na Slici 34.



Slika 34. Stranica za rukovanje korisničkim nalozima aplikacije

4.2.12 Stranica za rukovanje skriptovima

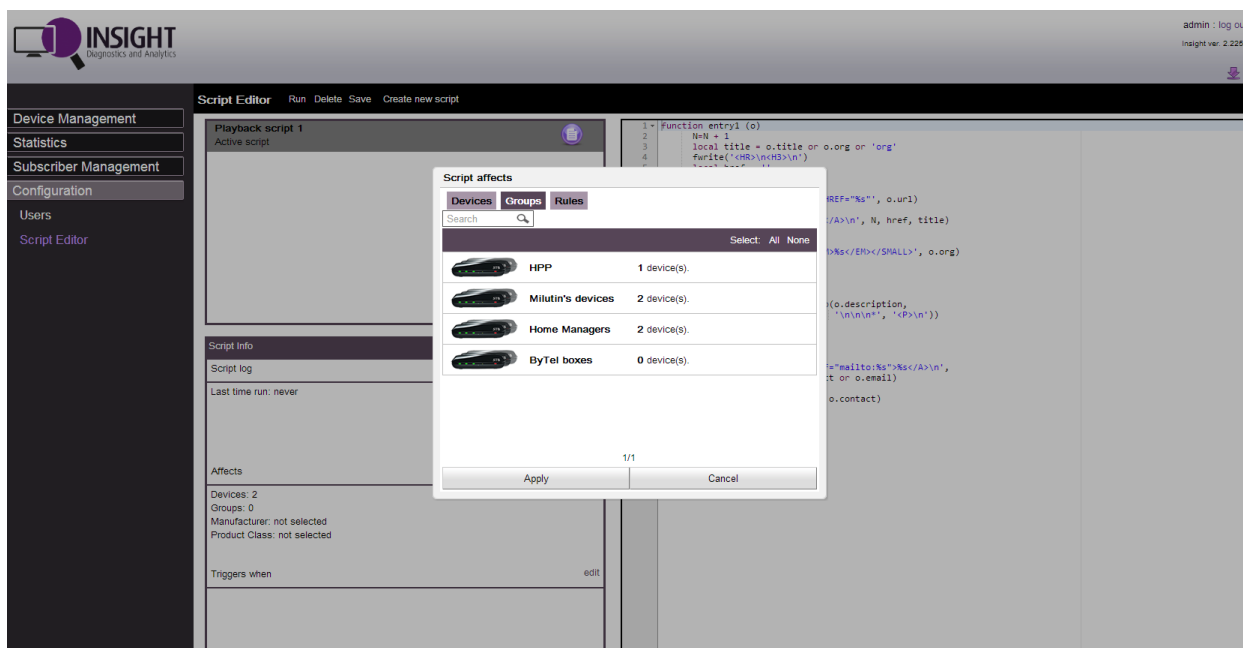
Koncept definisanja skriptova je važan kod realizacije autokonfiguracionog poslužioca. Korisnicima je data mogućnost pretplaćivanja na razne događaje radi izvršenja specifičnih operacija nad uređajima ili grupama uređaja. Kod realizovanog rešenja su podržani skriptovi pisani u Lua programskom jeziku[13]. Stranica je prikazana na Slici 35.



Slika 35. Stranica za rukovanje skriptovima

U desnom panelu se nalazi prozor za prikazivanje i izmenu skripte, koji podržava obeležavanje Lua izvornog koda pomoću *AceGWT* [14]biblioteke.

U gornjem levom panelu se nalazi spisak postojećih skriptova sa opcijama za kreiranje nove skripte i za brisanje, pokretanje i snimanje postojećih skriptova. Ispod toga se nalazi komponenta za podešavanje uslova za automatsko pokretanje skripte. Otvaranjem dijaloga (Slika 36) se mogu izabrati uređaji, grupe uređaja ili druga pravila za koje se izvršavanje skripte može vezati.



Slika 36. Dijalog za određivanje automatskog izvršenja skripte

4.3 Specifična rešenja u okviru aplikacije

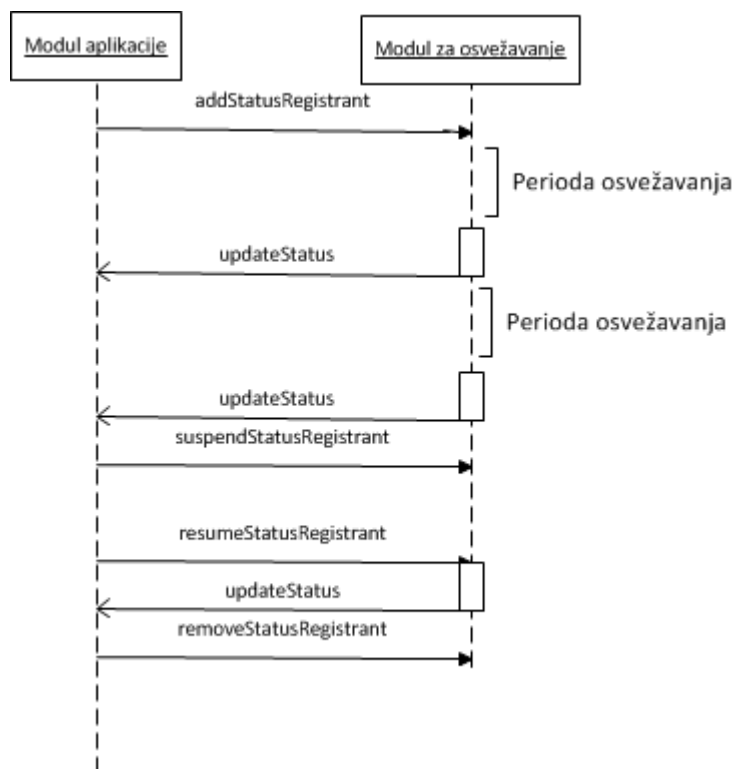
4.3.1 Osvežavanje modula aplikacije

Jedan od bitnih zahteva za savremene Web aplikacije je brzo i neprimetno osvežavanje sadržaja bez potrebe za interakcijom korisnika. Zato je potrebno da aplikacija ima jedan centralni modul koji periodično osvežava druge komponente koje to zahtevaju. Ulogu takvog modula u okviru ovog rešenja ima modul *StatusUpdater*. Ova klasa sadrži časovnik čiji se period podešava pomoću parametra pri registrovanju komponente za osvežavanje. Nakon isteka časovnika upoređuju se podaci sa klijentske strane i podaci sa strane poslužioca. Ako je došlo do promene neke vrednosti, poziva se metoda registrovanog *Presenter*-a za osvežavanje prikazanih podataka. Da bi se neki *Presenter* mogao registrovati za automatsko osvežavanje, potrebno je da se implementira Java interface *HasStatusChangeHandlers* koji obezbeđuje potrebne metode za osvežavanje. Unutrašnja logika *StatusUpdater* klase je tako projektovana da se u slučaju istovremenih isteka časovnika za osvežavanje ne pokreće izvršavanje zahteva ka poslužilacu

istovremeno, nego ih raspoređuje da se izvršavaju redom. Na ovaj način se pravi zaštita od zagušenja veze ka poslužilacu. Metode *StatusUpdater* klase su sledeće:

1. **publicvoid** `addStatusRegistrant (HasStatusChangeHandlers registrant, int intervalMillis)`
- Metoda služi za registraciju novog registranta. Ulazni parametri su komponenta koja treba da se periodično osvežava i perioda osvežavanja u milisekundama.
2. **publicboolean** `removeStatusRegistrant (HasStatusChangeHandlers registrant)`
- Metoda služi za uklanjanje registranta iz skupa registranata. Ulazni parametar je komponenta koja treba da se ukloni.
3. **publicboolean** `suspendStatusRegistrant (HasStatusChangeHandlers registrant)`
- Metoda služi za privremeno suspendovanje osvežavanja registranta. Ulazni parametar je komponenta koja treba da se suspenduje.
4. **publicboolean** `resumeStatusRegistrant (HasStatusChangeHandlers registrant)`
- Metoda služi za nastavak osvežavanja registranta. Ulazni parametar je komponenta koja treba ponovo da se osvežava.
5. **publicvoid** `suspendAllRegistrants ()`
- Metoda služi za privremeno zaustavljanje aktivnosti osvežavanja.

Primer komunikacije između modula aplikacije i klase *StatusUpdater* je prikazan na Slici 37.

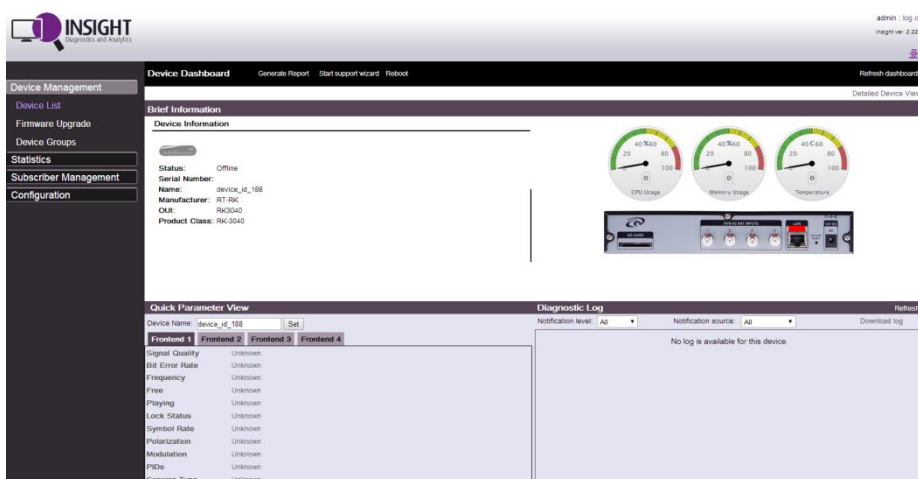


Slika 37. Primer komunikacije između modula aplikacije i klase *StatusUpdater*

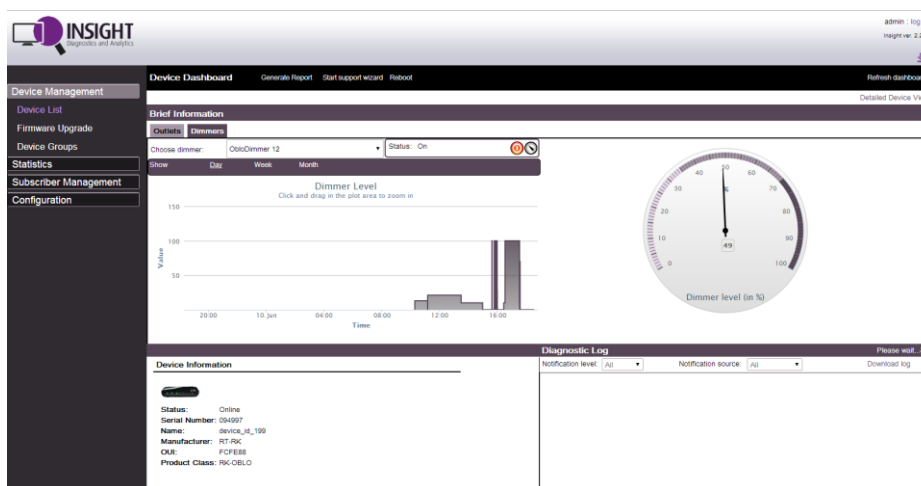
U ovakvim centralnim modulima aplikacije je najbitnije dobro rukovanje greškama jer se u slučaju otkazivanja neke operacije zaustavlja cela aplikacija.

4.3.2 Promenljiva stranica za praćenje statusa uređaja

U toku razvoja aplikacije je uočena potreba da za različite tipove uređaja budu dostupne različite stranice za pregled statusa uređaja. Ova potreba postoji zbog same prirode nadgledanih uređaja koji imaju različit skup parametara. Iz ove činjenice je nastala ideja za promenljivu stranicu za praćenje statusa uređaja. Primeri prilagođenih stranica se mogu videti na Slikama 38 i 39.

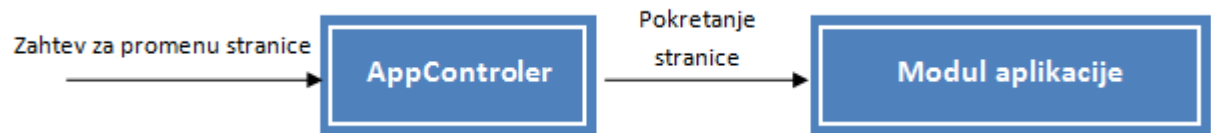


Slika 38. Stranica za praćenje statusa STB uređaja

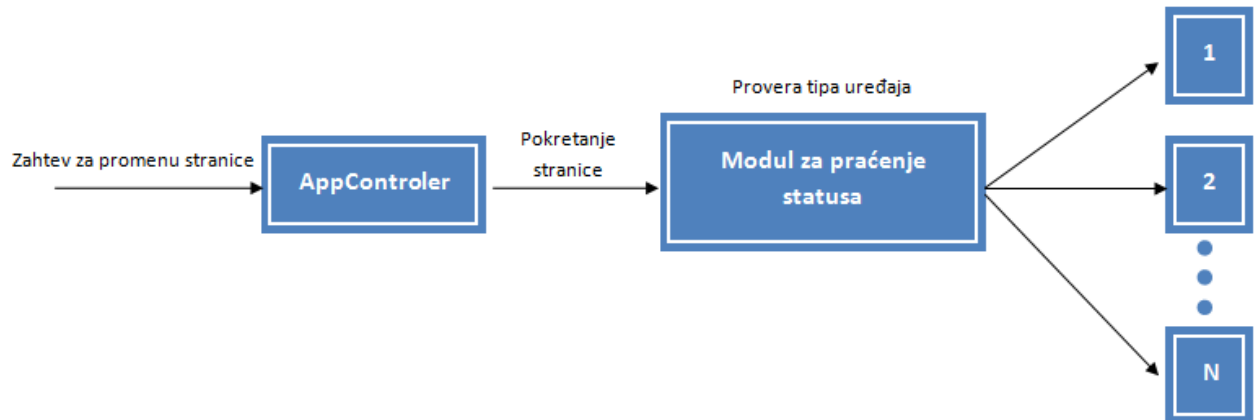


Slika 39. Stranica za praćenje statusa pametnih utičnica i prekidača

Da bi se ispunio ovaj zahtev, potrebno je implementirati dodatnu logiku za pokretanje stranice za praćenje statusa uređaja. Razlika između pokretanja modula za praćenje statusa i bilo kog drugog modula prikazuju slike 40 i 41.



Slika 40. Pokretanje druge stranice aplikacije



Slika 41. Pokretanje modula za praćenje statusa

5. Evaluacija rešenja

U ovom poglavlju data je evaluacija rešenja po sledećim kriterijumima:

1. Funkcionalnost i ispunjenost zahteva;
2. Vreme odziva;
3. Potrošnja sistemskih resursa.

5.1 Upoređivanje funkcionalnosti

Ispunjenost strukturnih zahteva je ključni zahtev evaluacije rešenja iz razloga što oni opisuju potrebe za stabilnošću i upotrebljivošću aplikacije. Analiza ispunjenosti ovih zahteva je prikazana u Tabeli 2.

	Status	Opis
Modularnost	✓	Rešenje je modularno, kao što je to opisano u poglavlju 3.2.
Prenosivost	✓	Obezbeđena korišćenjem GWT alata.
Upotrebljivost	✓	Izgled je dosledan i lak za korišćenje zbog upotrebe standardnih komponenti prilikom projektovanja.
Robustnost	✓	Robustnost je proverena kroz testiranje u realnim uslovima.

Tabela 2. Analiza ispunjenosti strukturnih zahteva

Upoređene funkcionalnosti po podržanim slučajevima korišćenja navedenim u poglavlju 3.1 su prikazani u Tabeli 3.

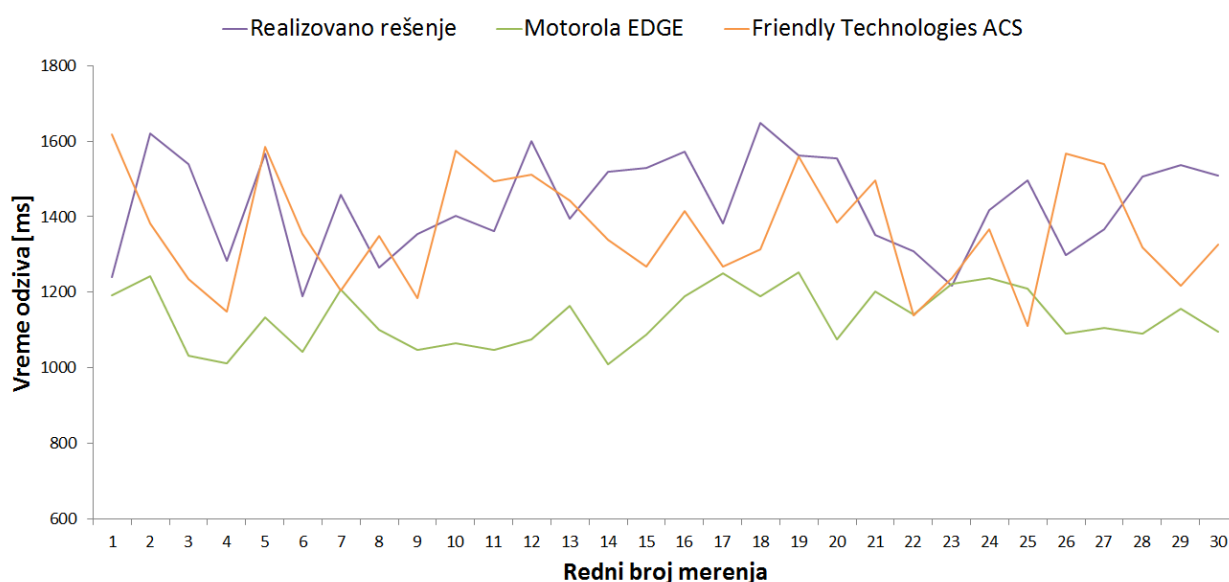
	Realizovano rešenje	Motorola EDGE	Friendly Technologies ACS
Pregled spiska uređaja	✓	✓	✓
Uvid u status uređaja	✓	✓	✓
Konfiguracija uređaja	✓	✓	✓
Rukovanje grupama uređaja	✓	✓	✓
Rukovanje korisničkim nalogima	✓	✓	✓
Rukovanje skriptovima	✓	✓	✓
Uvid u statistiku sistema	✓	✓	✓
Omogućen pristup korisnicima uređaja	✓	✗	✓
Prilagodljiv modul za prikazivanje statusa	✓	✗	✗

Tabela 3. Poređenje funkcionalnosti

Iz Tabele 3 se vidi da realizovano rešenje ne zaostaje za postojećim komercijalnim rešenjima u vidu funkcionalnosti. Uvedena je i nova funkcionalnost u vidu prilagodljivog modula za prikazivanje statusa uređaja.

5.2 Merenje vremena odziva

Pomoću alata *Firebug* [15] ispitivano je vreme odziva na komande iz aplikacije do prikaza rezultata na ekranu. Rezultati si prikazani na Slici 42.

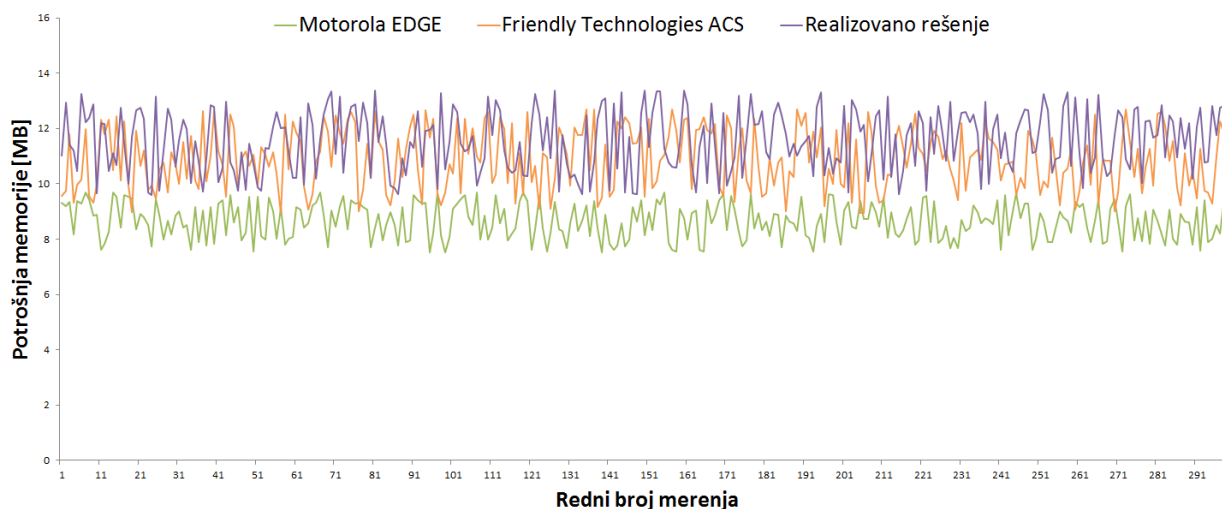


Slika 42. Rezultati merenja vremena odziva

Rezultati pokazuju da realizovano rešenje minimalno zaostaje za postojećim rešenjima u vidu vremena odziva, ali su izmerene vrednosti ispod preporučene maksimalne vrednosti za vreme odziva grafičkih aplikacija, koja iznosi 2 sekunde (2000ms) [16].

5.3 Potrošnja sistemskih resursa

Ova verifikacija je urađena u vidu merenja potrošnje memorije od Web aplikacije. Za ispitivanje je korišćeno razvojno okruženje internet pretraživača *Chrome*, koji omogućava automatizaciju pretraživača za svrhe ispitivanja. Ispitni slučaj sastoji se iz periodičnog menjanja stranice za prikaz liste uređaja, stranice za prikazivanje statusa uređaja i stranice za ažuriranje programske podrške. Ispitni slučaj sastoji se takođe od prolaska kroz liste i postavljanja parametara uređaja. Tokom ispitivanja zabeleženo je zauzeće memorije pretraživača. Prosečno zauzeće memorije je 11,5 MB, posle 300 izvršenih merenja. Grafikon sa rezultatima merenja je prikazan na Slici 43.



Slika 43. Rezultati merenjapotrošnje memorije

Rezultati pokazuju da realizovano rešenje troši približno istu količinu memorije kao i rešenje kompanije Friendly Technologies, dok malo zaostaje za rešenjem kompanije Motorola.

6. Zaključak

U radu je projektovana i implementirana korisnička aplikacija koja omogućava pristup funkcionalnostima poslužioca za automatsku konfiguraciju TR-069 uređaja. Aplikacija je realizovana tako da podržava širok spektar povezanih uređaja, krajnjih korisnika i poslužioca. Realizovano rešenje u potpunosti odgovara strukturnim i funkcionalnim zahtevima, a time su otvorene nove mogućnosti za dalja unapređenja i proširenja. Rešenje je testirano korišćenjem *InsightACS 2.0* konfiguracionog poslužioca. Verifikacija je pokazala da se realizovano rešenje može uporediti sa komercijalnim rešenjima što se tiče funkcionalnosti, dok u slučaju performansi postoji prostor za poboljšanja.

Fokus daljeg rada će biti unapređenje korisničke aplikacije u vidu novih funkcionalnosti, optimizacije potrošnje resursa i poboljšanja vremena odziva.

7. Literatura

- [1] H. Rachidi, „A framework for self-configuring devices using TR-069“, Int. Conf. on MCS, Morocco, 2011.
- [2] TR-069 (CPE Wan Management Protocol), Broadband Forum, www.broadband-forum.org/cwmp
- [3] TR-131 (ACS Northbound Interface Requirements), Broadband Forum, www.broadband-forum.org/technical/download/TR-131.pdf
- [4] Microsoft Application Architecture Guide, 2nd Edition, msdn.microsoft.com/en-us/library/ff650706.aspx
- [5] JQuery biblioteka za razvoj Web aplikacija, www.jquery.com
- [6] Google Web Toolkit, www.gwtproject.org/
- [7] Motorola EDGE™ Service Assurance Software Suite, [www.motorolasolutions.com/web/Business/_Documents/static files/EDGE Data Sheet 365-095-15888 x.1.pdf](http://www.motorolasolutions.com/web/Business/_Documents/static%20files/EDGE%20Data%20Sheet%20365-095-15888%20x.1.pdf)
- [8] Friendly Technologies ACS and TR-069 Management Console, www.friendly-tech.com/products/tr69-device-home-management/acs-and-tr-069-manage.aspx
- [9] MR-230 (TR-069 Deployment Scenarios), www.broadband-forum.org/marketing/download/mktgdocs/MR-230.pdf
- [10] Liu, L.; Watson, T.J.; Moulic, R.; Shea, D., "Cloud Service Portal for Mobile Device Management", ICEBE, Shanghai, pp. 474 - 478, 2010.
- [11] Google Maps SDK, developers.google.com/maps/
- [12] HighCharts biblioteka za grafički prikaz podataka, www.highcharts.com
- [13] Lua programski jezik, www.lua.org

-
- [14] AceGWT biblioteka za obeležavanje izvornog koda,
www.github.com/daveho/AceGWT
 - [15] Firebug alat za ispitivanje Web stranica, www.getfirebug.com
 - [16] Response time in man-computer conversational transactions, Miller, R. B., 1968.