

Osjeti ZG zrak Tehnička dokumentacija

Verzija 1.2

Sadržaj

1. Opis razvijenog proizvoda	3
1.1 Pregled problematike	3
1.2 Koncept sustava	3
1.3 Korištene tehnologije	4
2. Tehničke značajke	5
2.1 Osjetila i sklopovlje	5
2.1.1 Funkcijske cjeline.....	5
2.1.2 Sklopovlje za prihvata signala i kontrolu grijača.....	7
2.1.3 Mjerenje.....	8
2.1.4 Mjerenje temperature.....	9
2.1.5 Komunikacija.....	9
2.1.6 Sljedeći koraci.....	11
2.2 Aplikacija za pokretni telefon	11
2.2.1 Komunikacija pokretnog posrednika (Mobile Brokera) i senzora.....	12
2.2.2 Komunikacija mobilnog brokera (Mobile Broker) i Brokera.....	14
2.3 Statistika i baza podataka	17
2.3.1 Idejno rješenje komponente statistika.....	17
2.3.2 MySQL baza podataka.....	18
2.3.3 Poruke sustava Publish-Subscribe.....	19
2.3.4 Implementacija sustava.....	21
2.4 Web aplikacija	22
2.4.1 Korištenje podataka.....	23
2.4.2 Traka izbornika.....	24
2.4.3 Interaktivna karta i prikaz parametara mjerenja.....	25
2.4.4 Prikaz statistike za odabrano područje.....	27
2.4.5 Objavljivanje sadržaja na društvenoj mreži Facebook.....	28
Literatura	31

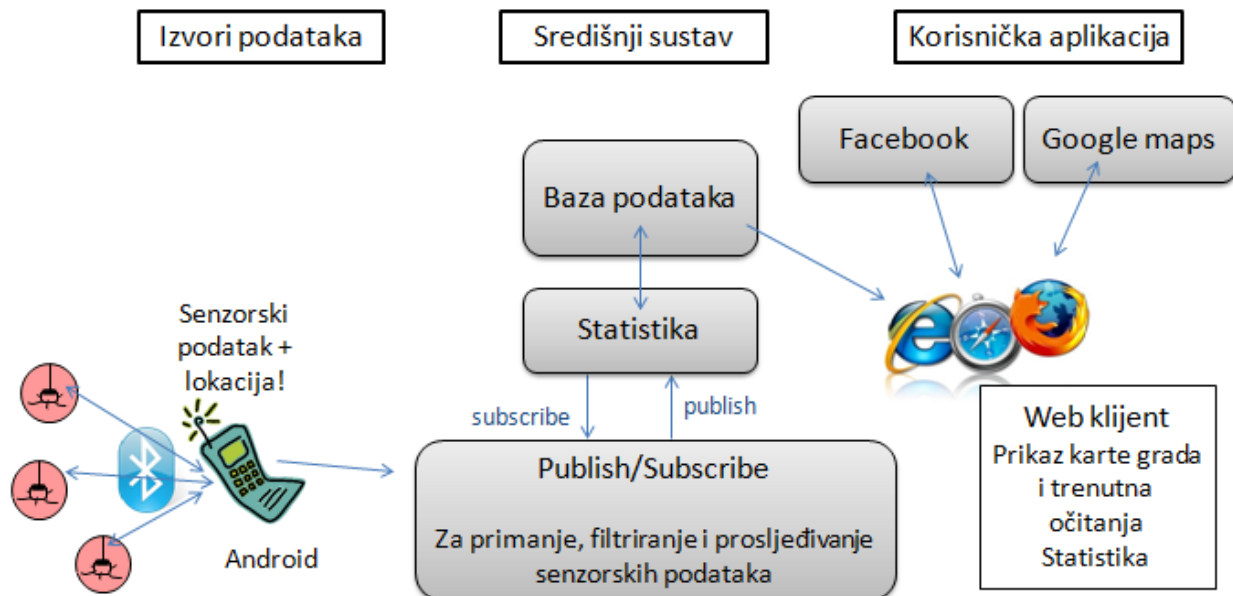
1. Opis razvijenog proizvoda

1.1 Pregled problematike

Svi želimo znati kakav zrak udišemo, a posebice kada nam o tome ovisi zdravstveno stanje. Osnovni zadatak našeg tima bio je razvoj pokretnog sustava za mjerenje kvalitete zraka. Postojeći sustav stacionarnih mjernih postaja nije na razini zadatka, jer mjeri kvalitetu zraka na unaprijed određenim lokacijama u gradu. Iako vrlo precizan, sustav stacionarnih mjernih postaja je skup, razmjerno velik što znači da treba osigurati zemljište za njegovo postavljanje te je samim tim ograničen na mali broj postaja koje mogu biti postavljene u gradu. Ako želimo dobiti korisniju i precizniju informaciju potrebno je podatke o kvaliteti zraka mjeriti na mnogo više lokacija.

1.2 Koncept sustava

U sklopu projekta izgrađen je prototipni informacijsko-komunikacijski sustav koji omogućuje mjerenje, prikupljanje, pohranjivanje i prikaz informacija o kakvoći zraka u gradu Zagrebu.



Slika 1: Konceptualni prikaz sustava

Osmišljeni sustav čine mali senzorski uređaj kojega korisnik nosi na odjeći ili torbi i pametni telefon s kojim je taj senzorski uređaj povezan i koji obrađuje i prosljeđuje informacije o kvaliteti zraka putem Interneta. Te se informacije ostalim korisnicima isporučuju putem specijalizirane internetske stranice. Dok se voze biciklom na posao, u školu ili na fakultet građani sami proizvode informacije kojima omogućuju sebi i ostalim zainteresiranim građanima, javno-zdravstvenom sustavu, gradskim vlastima i drugima objektivni uvid u kvalitetu zraka u gradu u stvarnom vremenu i na velikom broju lokacija, nadopunjujući tako postojeći sustav stacionarnih mjernih postaja.

1.3 Korištene tehnologije

U izradi projekta korištene su mnogobrojne tehnologije, a navedene su u nastavku.

- Osjetila i sklopovlje
 - razvojni alat TI EZ430-RF2560, koji sadrži mikrokontroler MSP430BT5190 i Bluetooth modul CC2560
- Aplikacija za pokretni telefon
 - Java
 - Android SDK
 - korišteni RESTful servisi
- Komponenta statistika
 - Java
- Baza podataka
 - MySql
- Web aplikacija
 - HTML, CSS
 - Apache poslužitelj
 - JavaScript, jQuery, jQueryUI
 - Google maps API, Google Chart API
 - Facebook JavaScript SDK

2. Tehničke značajke

Sustav se sastoji od četiri komponente. To su:

- komponenta za mjerenje kvalitete zraka, ostvarena sklopovljem i osjetilima za prikupljanje podataka u stvarnom vremenu;
- komponenta za prikupljanje i odašiljanje podataka, ostvarena aplikacijom za pokretni telefon;
- komponenta za prihvatanje, pohranu i obradu podataka, izvedena prema načelu *publish-subscribe*, ostvarena komponentama statistika i bazom podataka;
- komponenta za prikaz podataka korisnicima, ostvarena klijentskom web aplikacijom koja je povezana s Google maps i Facebook uslugama.

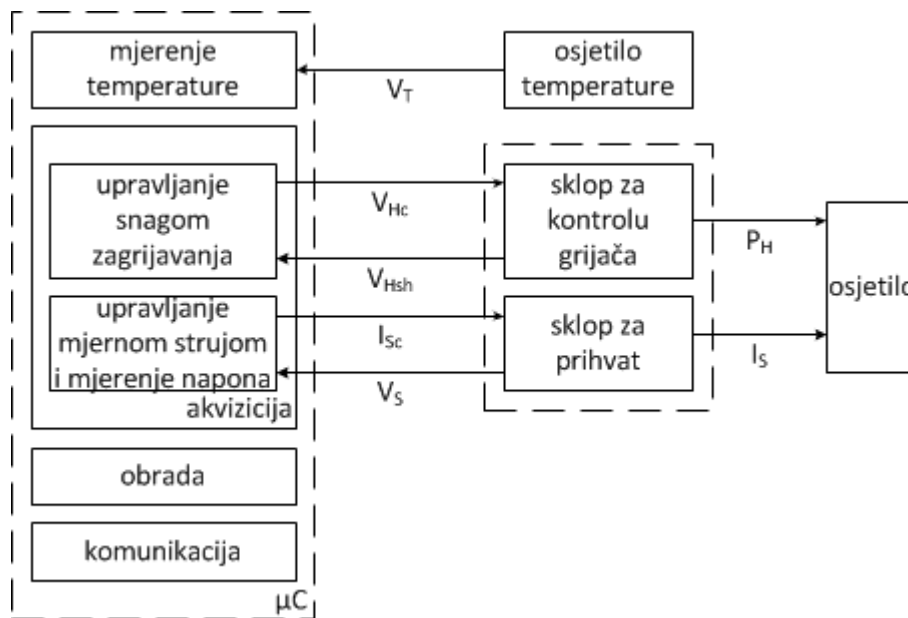
U nastavku poglavlja opisane su karakteristike i implementacija svake komponente te način komunikacije s ostalim komponentama sustava.

2.1 Osjetila i sklopovlje

2.1.1 Funkcijske cjeline

Mjerno sklopovlje se sastoji od sljedećih komponenti (slika 2):

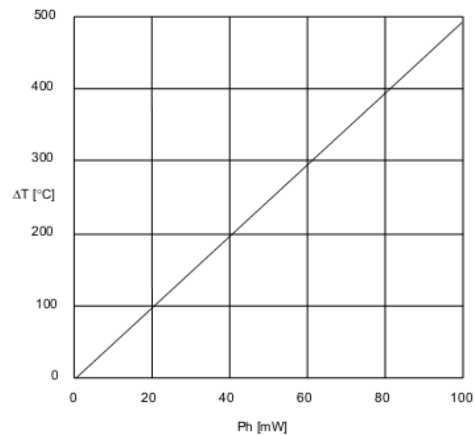
- osjetila koncentracije plina,
- osjetila temperature,
- sklopovlja za upravljanje i prilagodbu signala,
- mikrokontrolera.



Slika 2: Funkcijske cjeline mjernog sklopovlja

Radi jednostavnosti i preglednosti, prikazana je funkcionalnost za samo jedno osjetilo plina. Mikrokontroler ima zadaću upravljati pojedinim cjelinama, prikupljati mjerne podatke i obrađivati ih te komunicirati s pametnim telefonom putem Bluetootha. Za projekt se koristi razvojni alat TI EZ430-RF2560, koji sadrži mikrokontroler MSP430BT5190 i Bluetooth modul CC2560.

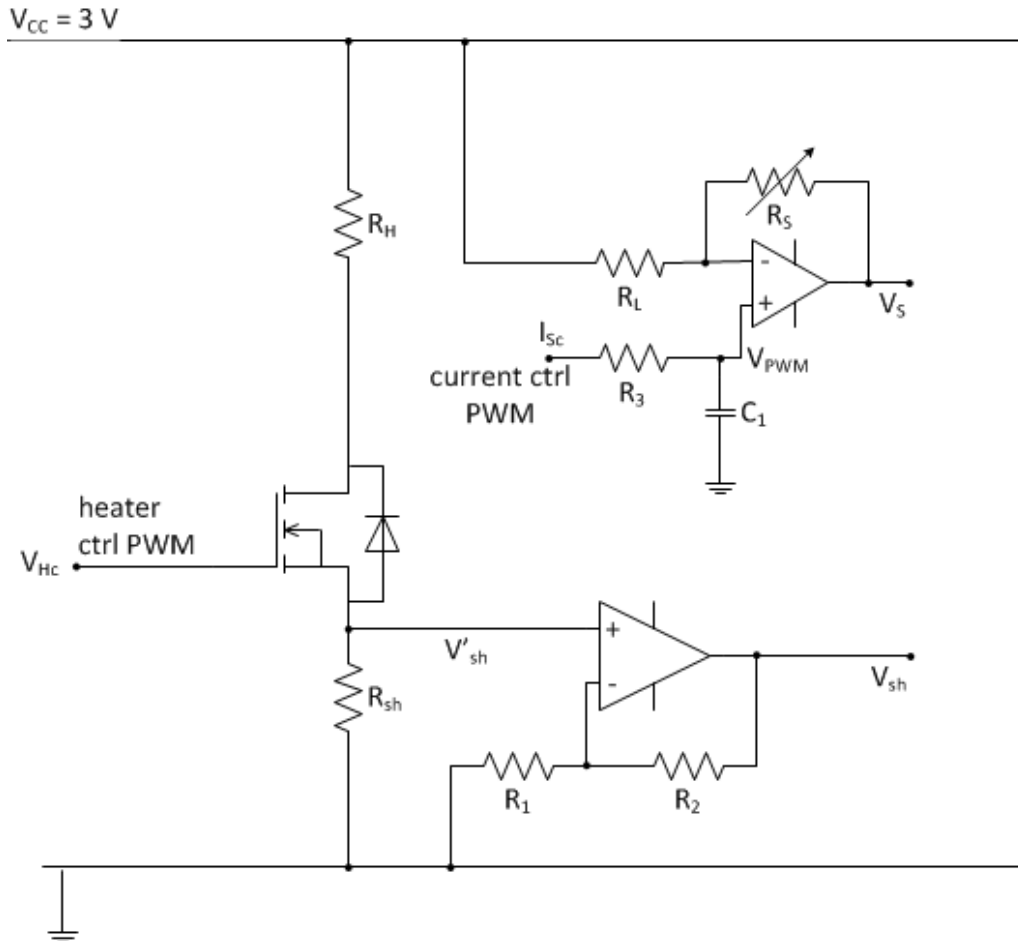
Za ispravno mjerenje, potrebna je kompenzacija temperature okoline, a informacija o njoj se dobiva iz ugrađenog osjetila temperature na korištenom modulu. Temperatura okoline određuje potrebnu snagu grijača P_H , koja je potrebna kako bi se postigla radna temperatura osjetila (slika 2). Snaga grijača se regulira signalom V_{HC} i povratnom vezom V_{Hsh} .



Slika 3: Ovisnost razlike temperature osjetni sloj - okolina o snazi grijača

Kako bi se izmjerio otpor osjetnog sloja, mjeri se pad napona na njemu i izračunava uz poznati podatak o struji koja prolazi njime. Mikrokontroler upravlja strujom osjetnog sloja signalom I_{sc} , a mjeri napon V_s . Iz izmjerenog napona, izračunava se otpor osjetnog sloja iz kojeg se, prema specifikacijama proizvođača, određuje koncentracija ciljanog plina.

2.1.2 Sklopovlje za prihvat signala i kontrolu grijača



Slika 4: Shema mjernog sklopovlja

Za napajanje sklopa (Slika 4) je predviđeno baterijsko napajanje napona $V_{CC} = 3\text{ V}$. Kako bi se osiguralo točnije mjerenje, napon napajanja se mjeri u trenutku akvizicije napona sa osjetila.

Komponenta osjetila se sastoji od dva dijela: osjetilnog sloja R_S i grijača R_H (slika 3). Grijač R_H je disipacijski otpornik, koji služi za zagrijavanje promjenjivog otpornika R_S .

Za kontrolu grijača se koristi PWM signal s mikrokontrolera V_{Hc} , koji spojen na gate elektrodu MOSFET-a periodički uključuje i isključuje tranzistor, u ovom slučaju sklopku. Signal je frekvencije 10 kHz, a njegov djelatni ciklus D određuje snagu zagrijavanja:

$$P_H = \frac{V'_{sh}}{R_{sh}} \sqrt{D} (V_{CC} - V'_{sh} \cdot \sqrt{D}),$$

gdje je,

$$V'_{sh} = \frac{R_1}{R_1 + R_2} V_{sh},$$

a V_{sh} izmjereni napon na shuntu R_{sh} , doveden na ADC mikrokontrolera kao povratna veza regulacije snage grijača. Vrijednost shunta R_{sh} ovisi o maksimalnoj dozvoljenoj disipaciji pojedinog osjetila i predviđenog otpora disipacijskog otpornika (tabela 1).

Tabela 1: Veličine otpornika Rsh

osjetilo	P_{Hmax}/mW	R_{Hmax}/Ω	$R_{sh\ min}/\Omega$
MiCS-5525	81	82	13,45
MiCS-5121	81	82	13,45
MiCS-2710	50	73	41,63
MiCS-2610	95	78	7,96

Uzevši u obzir ulazni opseg ADC-a, otpori R_1 i R_2 bi trebali biti jednaki, tako da pojačanje iznosi dva. Promjenjivi otpornik R_s predstavlja otpor osjetnog sloja, koji se mijenja s koncentracijom plina u zraku. Struja kroz taj otpornik je poznata:

$$I_S = \frac{V_{CC} - V_{PWM}}{R_L}$$

V_{PWM} služi za regulaciju struje kroz R_s i za dovođenje signala unutar mjernog opsega ADC-a. Otpornik R_L se određuje s obzirom na maksimalnu dozvoljenu disipaciju osjetnog sloja, kako ne bi došlo do oštećenja (tablica 2). R_0 je otpor osjetnog sloja u zraku.

Tabela 2: Veličine otpornika RL

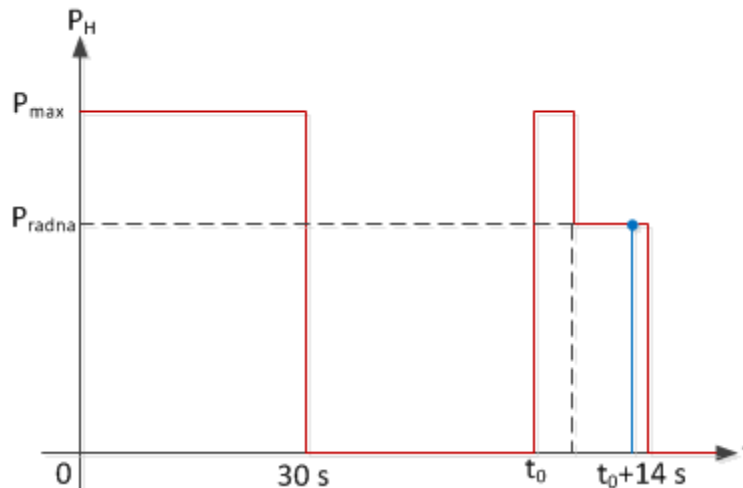
osjetilo	P_{Smax}/mW	$R_{0max}/k\Omega$	$R_{L\ min}/k\Omega$
MiCS-5525	8	100	10,6
MiCS-5121	8	100	10,6
MiCS-2710	1	0,8	2,7
MiCS-2610	1	3	5,2

Otpor osjetnog sloja se izračunava prema:

$$R_S = R_L \frac{V_{PWM} - V_S}{V_{CC} - V_{PWM}}$$

2.1.3 Mjerenje

Ciklus mjerenja se sastoji od nekoliko dijelova. Pri uključivanju sklopovlja, pokreće se inicijalizacija osjetila (slika 4). Kako su osjetila bila neaktivna za vrijeme dok je sklopovlje isključeno i dok se ne mjeri, moguće je nakupljanje stranih tvari i vlage na osjetnom sloju. Kako bi se osjetni sloj očistio, potrebno ga je grijati maksimalnom dozvoljenom snagom tijekom 30 sekundi. Kako postoje 4 osjetila plina, proces inicijalizacije traje 2 minute, budući da istovremeno ne može raditi više od jednog osjetila.



Slika 5: Ciklus zagrijavanja, $t = 0$ uključivanje, $t = t_0$ početak mjerenja

Osjetilo je u stanju čekanja dok ne dobije nalog za mjerenje od strane telefona. Ciklus mjerenja traje 15 sekundi po jednom osjetilu. Prvih 5 sekundi je predgrijavanje osjetnog sloja maksimalnom snagom, kako bi ga se očistilo od eventualnih nečistoća. Sljedećih 10 sekundi se osjetni sloj grije snagom da se postigne radna temperatura osjetnog sloja. Mjerenje se radi pred kraj ciklusa, u 14 sekundi ($t_0 + 14$, slika 4), jer je potrebno da se uspostavi kemijska ravnoteža na osjetnom sloju.

2.1.4 Mjerenje temperature

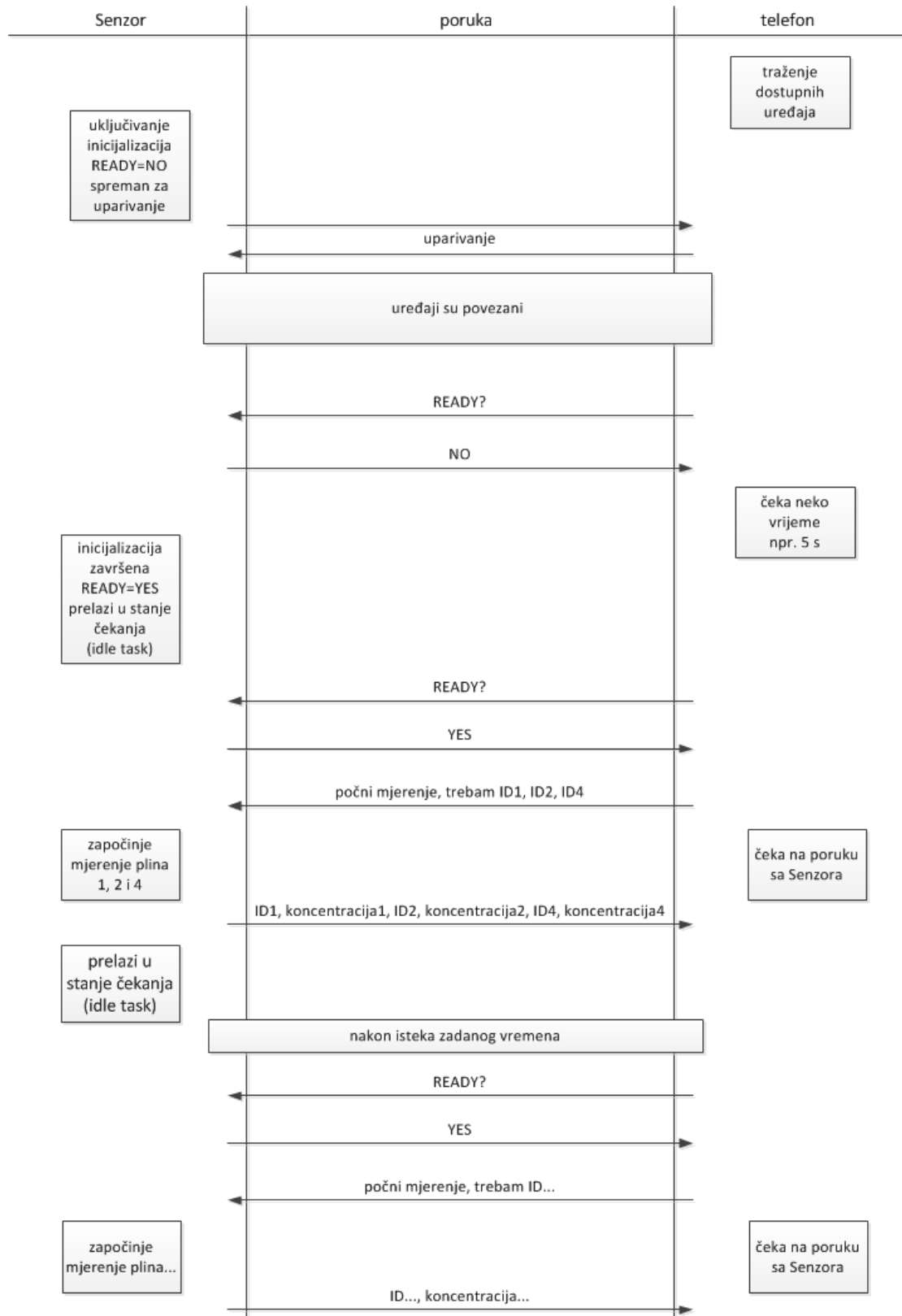
Razvojni alat EZ430-RF2560 sadrži RF2560T pločicu na kojoj se nalaze mikrokontroler, Bluetooth modul i osjetilo temperature. Korištenjem i modifikacijom demo aplikacije razvojnog alata, napravljena je aplikacija koja putem I²C sabirnice čita vrijednost iz osjetila temperature, izračunava stvarnu temperaturu i šalje je preko Bluetootha na pametni telefon. Poruka koja se šalje se sastoji od 3 ASCII znaka: predznak temperature i dvije znamenke za vrijednost. Kako osjetilo temperature radi u području od -40°C do $+85^{\circ}\text{C}$, dvije znamenke su dovoljne za vrijednost temperature.

Nije pisana posebna aplikacija za pametni telefon, već je korišten Bluetooth terminal, kako bi se vidjeli podaci koje je telefon primio.

2.1.5 Komunikacija

Sklopovlje osjetila komunicira s pametnim telefonom preko Bluetooth veze (slika 6). Aplikacija na telefonu određuje plinove čije će se koncentracije mjeriti. Nakon uključivanja osjetila i telefona, spremni su za povezivanje. Iako je veza uspostavljena, osjetilo još nije spremno za rad, jer mora napraviti inicijalizaciju uređaja. Iz tog razloga, ali i kao sigurnosna mjera, uvodi se kontrolna varijabla `READY`.

Ako je `READY = YES`, znači da je uređaj spreman za primanje instrukcija i mjerenje, a ukoliko je `READY = NO` uređaj je još u stanju inicijalizacije ili se dogodila neka greška u radu. Za vrijeme dok je `READY = NO` i postoji potreba za mjerenjem, telefon periodički ispituje stanje spremnosti (slika 5). Međutim, ako osjetilo ne dojavi spremnost unutar jedne minute, treba pretpostaviti pogrešku i smatrati osjetilo trenutačno nesposobnim za rad.



Slika 6: Slijed Bluetooth komunikacije osjetila i telefona

Nakon što je osjetilo telefonu dojavilo svoju spremnost, telefon šalje listu identifikatora plinova (ID) čije je koncentracije potrebno izmjeriti (tabela 3). Nakon toga počinje proces mjerenja, čije trajanje ovisi o broju traženih koncentracija.

Tabela 3: Pregled poruka koje se šalju Bluetooth vezom i njihove veličine

Poruka	Oblik	Veličina [byte]
ID osjetila temperature	0000 0000	1
ID osjetila plina 1	0000 0001	1
ID osjetila plina 2	0000 0010	1
ID osjetila plina 3	0000 0011	1
ID osjetila plina 4	0000 0100	1
READY (YES)	0111 0000	1
READY (NO)	0100 0000	1
Koncentracija plina [ppm]	-	2

Kada su mjerenja gotova, stvara se poruka oblika:

ID1	koncentracija1	ID2	koncentracija2	ID4	koncentracija4
-----	----------------	-----	----------------	-----	----------------

koja je, za ovaj primjer kada su tražene koncentracije plinova 1, 2 i 4, duljine:

$$3 \cdot 8 + 3 \cdot 16 = 72 \text{ bita.}$$

Time je zaključena izmjena poruka za taj ciklus mjerenja. Osjetilo prelazi u stanje čekanja i izvršava idle task dok mu sa Bluetooth veze ne dođu novi zahtjevi.

Sklopovlje osjetila je u stanju čekanja cijelo vrijeme dok postoji veza s pametnim telefonom. Pametni telefon je taj koji prekida vezu, nakon čega osjetilo ponovo postaje slobodno za povezivanje s nekim telefonom, dok mu se ne isključi napajanje.

2.1.6 Sljedeći koraci

- nabava potrebnih komponenti za sklopovlje i proizvodnja sklopovlja,
- implementacija programske funkcionalnosti mjerenja i regulacije zagrijavanja.

2.2 Aplikacija za pokretni telefon

Aplikacija AirSense za pokretni telefon namijenjena je za prikupljanje, obradu i filtriranje informacija o kakvoći zraka. Podaci se očitavaju sa senzora i korištenjem Bluetooth tehnologije prenose do mobilne aplikacije radio prijenosom. Podaci koji se prikupljanju su:

- koncentracija plina ugljikov monoksid (CO),
- koncentracija plina dušikov dioksid (NO₂),
- koncentracija plina ozon (O₃),
- i temperatura.

Komunikacija između komponente senzor i pokretne aplikacije opisana je u potpoglavlju 2.2.1. U prvoj verziji aplikacije podaci o kakvoći zraka ne dobivaju se sa senzora, već korištenjem simulatora „Sensor

simulator“ za simuliranje okoline na operacijskom sustavu Android, razvijenog u sklopu *Diplomskog rada Josipa Balića*. Vrijednosti podataka također se mogu simulirati čitanjem iz datoteke pohranjene na disku.

Izvorni podaci prikupljeni sa senzora prosljeđuju se do posrednika, *Brokera*, čiji se rad temelji na modelu objavi-pretplati. *Broker* prosljeđuje podatke do komponente *Statistika* na temelju definiranih aktivnih pretplata od strane *Statistike*. Komunikacija između mobilne aplikacije i sustava objavi-pretplati te komunikacija između komponenti same mobilne aplikacije objašnjena je u potpoglavlju 2.2.2.

2.2.1 Komunikacija pokretnog posrednika (*Mobile Brokera*) i senzora

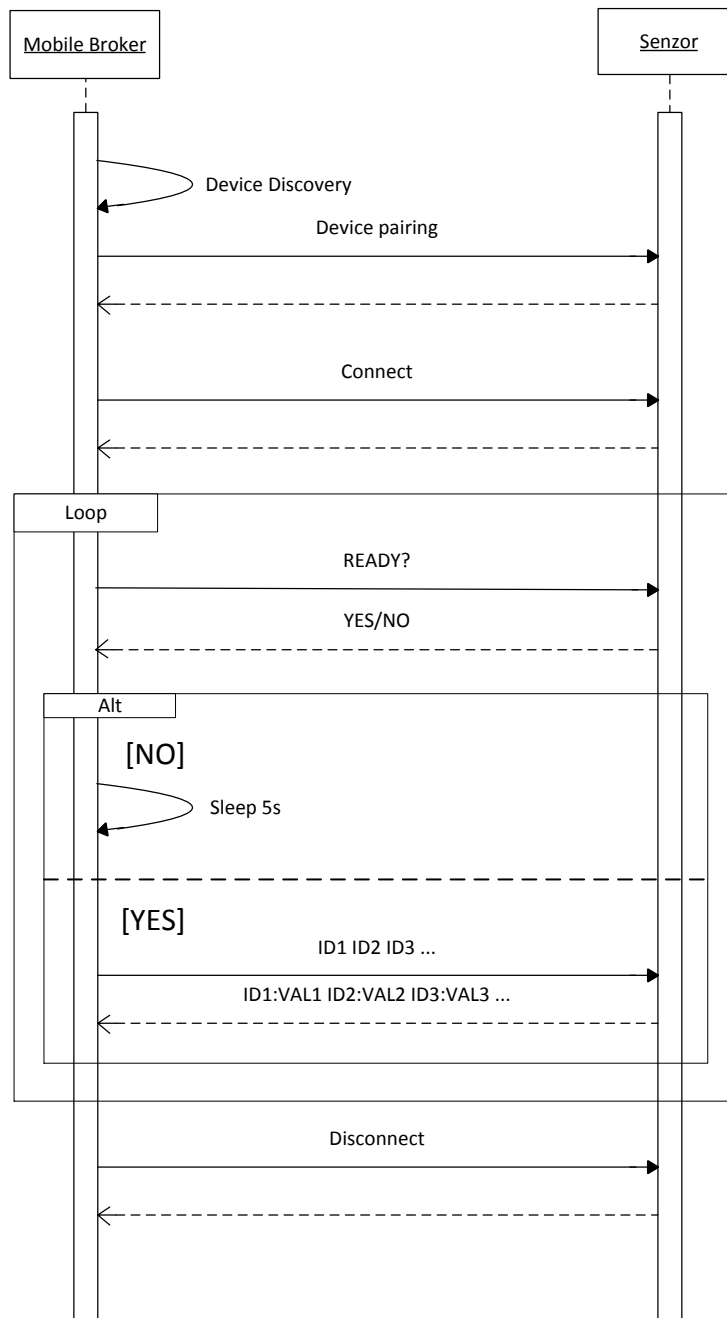
Prilikom korištenja AirSense aplikacije za pokretne uređaje, moguće je postaviti da aplikacija radi kao objavljivač obavijesti. Za omogućavanje ove funkcionalnosti potrebno je povezati pokretni uređaj sa sensorom. Pametni telefon i senzor se u neposrednoj blizini (za većinu pokretnih uređaja do 10m) povezuju tehnologijom Bluetooth i ostvaruju povezanost u PAN (engl. *Personal Area Network*) mreži. Uređaji komuniciraju radio prijenosom, komunikacija je opisana u nastavku, te prikazana na slici 7.

Prije uspostave veze između mobilne aplikacije i senzora, prvo je potrebno otkriti prisutnost senzora. Pozivom metode `startDiscovery()` mobilne aplikacije vrši se pretraživanje okoline za uređajima koji imaju omogućen Bluetooth, te kojima je omogućeno otkrivanje. Za svaki novootkriveni uređaj korisniku se nudi mogućnost uparivanja uređaja. Prilikom uparivanja, uređaji razmjenjuju osnovne informacije kao što je ime uređaja, klasa uređaja i MAC adresa uređaja. Korištenjem prikupljenih informacija uređaji mogu uspostaviti vezu.

Prije uspostave konekcije s sensorom, pokretna aplikacija dohvaća vrata putem kojih će komunicirati s sensorom tako da protokolom SDP (engl. *Service Discovery Protocol*) provodi *lookup* za predani UUID (engl. *Universally Unique Identifier*), koji mora odgovarati UUID-u senzora. Kao UUID za uređaje sa Serial Port profilom se koristi dobro poznata vrijednost:

0001101 – 0000 – 0000 – 8000 – 00805F9B34FB.

Kada nam je port poznat poziva se metoda `connect()`, SDP protokolom se za zadani UUID provodi *lookup* za RFCOMM (engl. *Serial Port Emulation/Radio Frequency Communication*) kanal koji će se koristiti za komunikaciju. Ukoliko je pretraga uspješna, te ukoliko senzor prihvati konekciju, metoda `connect()` će uspješno završiti, u suprotnom će aplikacija baciti iznimku. Uređaji se smatraju povezanim ukoliko komuniciraju putem istog RFCOMM kanala. Nakon povezivanja se dohvaćaju ulazni i izlazni tokovi za komunikaciju za sensorom.



Slika 7: Komunikacija mobilnog brokera (*Mobile Broker*) i senzora

Nakon povezivanja uređaja mobilna aplikacija može senzoru slati zahtjeve za mjerenjima. Prije svakog zahtjeva za podacima, mobilna aplikacija će porukom `READY?` upitati senzor da li je spreman za slanje podataka. Provjerava se spremnost senzora kako bi se izbjegla pogrešna mjerenja u početku rada senzora ili u slučaju greške tijekom rada senzora. Ukoliko je senzor spreman za slanje podataka mobilnoj aplikaciji će vratiti vrijednost 112 (`YES`) ili u suprotnome vrijednost 64 (`NO`).

U slučaju da mobilna aplikacija primi odgovor `NO`, čekati će unaprijed definirano vrijeme (npr. 5s).

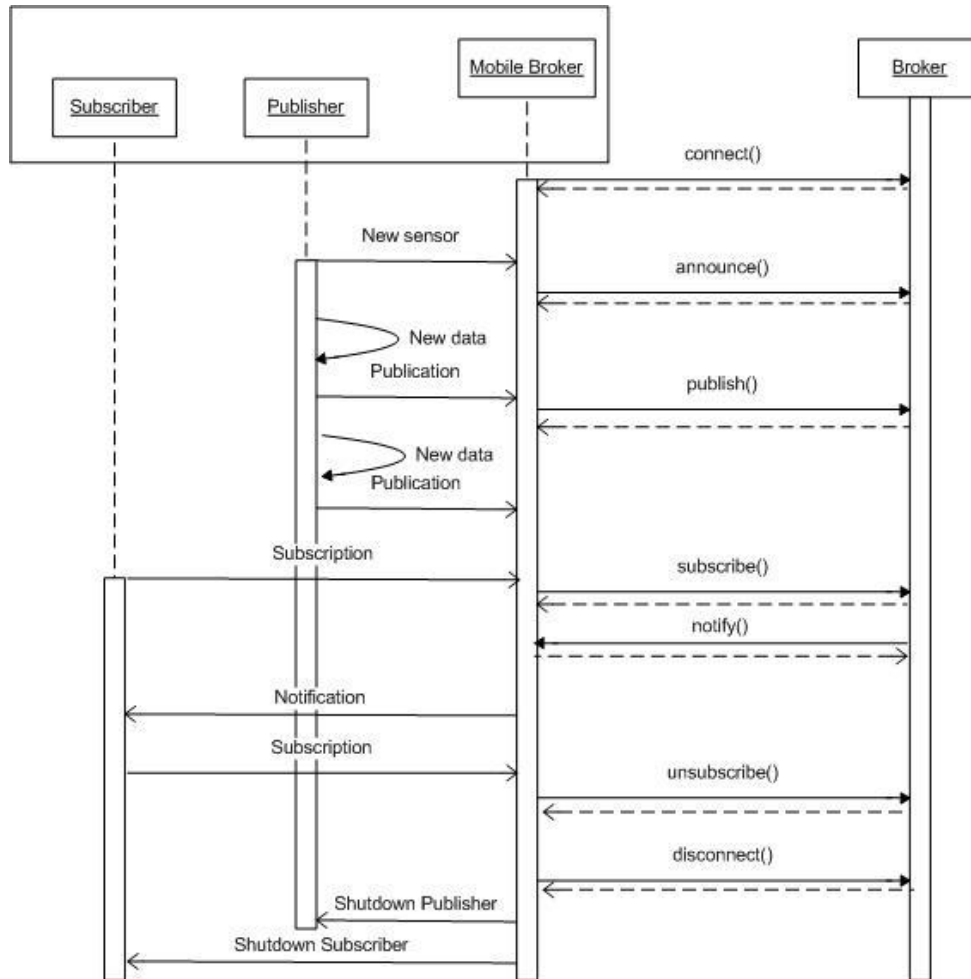
Ukoliko senzor odgovori porukom `YES`, mobilna aplikacija šalje zahtjev senzoru sa identifikatorima osjetila sa kojih ga zanima mjerenje. Format poruke je slijed bajtova, gdje svaki bajt sadrži identifikator osjetila. Identifikator osjetila je cijeli broj. Nakon primanja poruke, senzor obavlja mjerenja na traženim osjetilima. Proces mjerenja traje otprilike 15 sekundi po osjetilu. Nakon mjerenja, senzor šalje mobilnoj aplikaciji poruku sa rezultatima mjerenja. Poruka je slijed parova ID osjetila i izmjerena vrijednost na senzoru (1 bajt ID, 2 bajta izmjerena vrijednost).

Kada aplikacija želi prekinuti komunikaciju sa senzorom, poziva metodu `closeConn()`. Pozivom metode se zatvaraju ulazni i izlazni tokovi, te port preko kojih aplikacija komunicira sa senzorom.

2.2.2 Komunikacija mobilnog brokera (*Mobile Broker*) i Brokera

AirSense aplikacija namijenjena je za izvođene na pametnim telefonima s operacijskim sustavom Android. Glavna komponenta aplikacije je tzv. pokretni posrednik, odnosno, *Mobile Broker*, koji u sebi sadrži entitete: objavljivač podataka (*Publisher*) te pretplatnik (*Subscriber*). *Mobile Broker* komunicira s entitetima *Publisher* i *Subscriber*. Entiteti *Publisher* i *Subscriber* su zasebni procesi koje nadzire proces *Mobile Broker*. Ta komunikacije izvedena je interno te nije vidljiva izvan sustava i prikazana je na Slici 8.

Unutar aplikacije navedeni procesi izvedeni su kao tzv. servisi (engl. *Services*). *Service* je komponenta aplikacije izvedena za Android operacijski sustav koja može u pozadini dugotrajno izvoditi operacije. Svakoj komponenti *Subscriber* i *Publisher* pridružena je jedna komponenta *Activity* koja osigurava korisničko sučelje preko kojeg korisnik može zatražiti mjerenje određenih podataka (*Publisher* komponenta) ili može zatražiti prikaz podataka o kakvoći zraka za željeno odabrano područje unutar grada (*Subscriber* komponenta). *Mobile Broker* također komunicira s posrednikom (*Broker*), čiji rad se temelji na modelu objavi-pretplati. Ta komunikacija izvedena je putem REST protokola, odnosno RESTful web usluge. Ideja je da se svaki resurs, u ovom slučaju, npr. pretplata ili obavijest, dohvati, postavi ili obriše jednom od metoda protokola HTTP (Get, Post, Put ili Delete) sa nekog zadanog URI-a u XML ili JSON formatu. Na ovaj način osigurava se da su potrebni resursi uvijek dostupni te se osigurava razmjerni rast (skalabilnost) jer su RESTful web usluge tipično bez stanja i koriste priručni spremnik.



Slika 8: Komunikacija između entiteta Mobile Broker, Broker, Publisher i Subscriber

Prilikom pokretanja aplikacije *Mobile Broker* komponenta se prijavljuje na sustav objavi-pretplati (posrednik *Broker*) metodom `connect()`. Metodom `connect()` šalju se informacije o čvoru (IP adresa, port i ime *Mobile Broker-a*,) te trajanje komunikacije (*expiry time*). Ukoliko se komponenta *Mobile Broker* odjavljuje sa sustava, šalje se vrijednost za trajanja komunikacije 0. Metoda `connect()` koristi HTTP metodu PUT za stvaranje novog ili zamjenu postojećeg elementa koji je adresiran zadanim URI-em, npr.:

HTTP PUT <http://www.AirSense.com/users/{useruri}>

U tijelu HTTP poruke šalju se gore navedene informacije o čvoru te trajanje komunikacije. Kao odgovor, od *Brokera* prema *Mobile Brokeru* šalje se HTTP odgovor sa statusom 200 OK ako je poslužitelj (*Broker*) primio, razumio i ispunio zahtjev ili 400 Bad Request ukoliko je zahtjev neispravan. U slučaju odjave *Mobilnog Brokera* sa *Brokera* koristi se metoda `disconnect()`, odnosno HTTP metoda DELETE kojom se briše resurs (`/users/{useruri}`).

Nakon što entitet *Publisher* otkrije novi senzor (definirano u potpoglavlju 2.2.1), dojavljuje to *Mobile Brokeru* koji šalje metodom `announce()` detalje o senzorima *Brokeru*. Protokolom HTTP i metodom

POST dohvaćaju se pretplate koje se odnose na mobilni čvor, odnosno na podatke koje senzori s kojima je mobilna aplikacija povezana mogu mjeriti. U tijelu poruke šalju se podaci o senzovima, kao što je identifikator senzora prema kojem se točno zna koji podatak (koncentraciju plina, temperaturu) taj senzor može mjeriti:

```
HTTP GET http://www.AirSense.com/users/{useruri}/subscription/
```

Kao odgovor na metodu `announce()`, prima se odgovor HTTP protokolom sa statusom o uspjehu ili neispravnosti zahtjeva. U tijelu odgovora nalaze se pretplate koje „prekrivaju“ poslane parametre senzora.

Korisnik putem korisničkog sučelja entiteta *Publisher* može zatražiti mjerenje određenih koncentracija plinova ili temperature, što *Publisher* zatim traži od senzora. *Publisher* mjerenja šalje do *Mobile Brokera* koji, nakon primitka poruke od strane *Publishera*, dohvaća informacije o svojoj lokaciji korištenjem GPS-a. Usporedbom informacija o geografskoj širini i visini dobivenih od GPS-a i istih vrijednosti za sva definirana područja mjerenja, *Mobile Broker* može znati oznaku područja u kojem je mjerenje izvršeno. Ukoliko dolazi do preklapanja pristiglih obavijesti prije dobivenim pretplatama, *Mobile Broker* vrši objavu podataka `publish()` porukom. U slučaju da se očitaju podaci za koje nema definiranih pretplata, *Mobile Broker* ne šalje ništa. Objava se vrši metodom PUT protokola HTTP:

```
HTTP PUT http://www.AirSense.com/users/{useruri}/publication/
```

U tijelu poruke šalju se strukturirane objave (u obliku *atribut, operator, vrijednost*) u XML formatu. Odgovor se šalje protokolom HTTP na već opisane načine, a u tijelu odgovora pristiže identifikator objave.

U slučaju da se iz nekog valjanog razloga objava želi obrisati sa *Brokera*, *Mobile Broker* šalje poruku `unpublish()` metodom DELETE:

```
HTTP DELETE  
http://www.AirSense.com/users/{useruri}/publication/{pubID}
```

Kao odgovor dobiva informaciju o uspjehu ili neuspjehu zahtjeva.

Osim što može zatražiti mjerenje koncentracije plinova i temperature zraka za trenutno područje u kojem se nalazi u stvarnom vremenu, korisnik može zatražiti prikaz informacija o kakvoći zraka za bilo koje područje u nekom vremenu. Korisnik odabire koje informacije želi da mu se prikažu na zaslonu korištenjem korisničkog sučelja entiteta *Subscriber*. Pritom se korisniku nude informacije o podacima koji se mjere kao i područja unutar kojih se mjerenja mogu vršiti. Entitet *Subscriber* zatim formira pretplatu u XML formatu i prosljeđuje tako definiranu pretplatu do *Mobile Brokera*. Pretplata se do Brokera prenosi metodom `subscribe()` protokolom HTTP:

```
HTTP POST http://www.AirSense.com/users/{useruri}/subscriptions
```

U tijelu poruke prenosi se detalji o pretplati, odnosno informacije koje je zatražio korisnik te trajanje poruke, koje može biti 0, u slučaju da pretplata više nije važeća. U odgovoru *Brokera* šalje se identifikator pretplate, naravno uz uspjeh zahtjeva ili informaciju o grešci na klijentu.

Ukoliko se pretplata želi povući sa *Brokera*, znači da korisnika više ne zanimaju informacije definirane u pretplati, *Mobile Broker* šalje poruku `unsubscribe()`. Koristi se metoda DELETE protokola HTTP:

```
HTTP DELETE  
http://www.AirSense.com/users/{useruri}/subscription/{subID}
```


U slučaju da na *Brokeru* postoje definirane obavijesti koje su prekrivene pretplatama definiranim od strane korisnika, *Broker* šalje poruku `notify()`:

HTTP POST <http://www.AirSense.com/users/{useruri}/publications>

Mobile Broker šalje odgovor *Brokeru* u kojem se šalje informacija o uspješnosti zahtjeva. Poruka u svojem tijelu sadrži obavijesti u XML formatu, koje *Mobile Broker* prosljeđuje *Subscriberu*. Entitet *Subscriber* informacije dobivene u pretplati prikazuje korisniku putem korisničkog sučelja.

Prilikom prestanka rada aplikacije *Mobile Broker* šalje *Brokeru* `disconnect()` i zaustavlja procese *Publisher* i *Subscriber*.

2.3 Statistika i baza podataka

Statistika jest komponenta koja vrši prijenos podataka s pametnih telefona u MySQL bazu podataka i računa srednje vrijednosti za unaprijed definirane vremenske intervale. Za komunikaciju aplikacije s pametnim telefonima se koristi gotov programski paket objava i pretplata (*publish-subscribe*). Sustav *publish-subscribe* se sastoji od skupa pretplatnika, skupa objavljiivača i komponente *broker* koja je zadužena za preusmjeravanje poruka unutar sustava. Podaci s pametnih telefona se objavljuju porukom „*publish*“ u *publish-subscribe* sustav i privremeno pohranjuju na *broker* komponenti. Pretplatnik, u ovom slučaju komponenta *statistika*, se pretplaćuje porukama „*subscribe*“ na komponentu *broker* i na taj način prikuplja podatke koje zapisuje u MySQL bazu.

U komponenti *statistika* je definiran vremenski interval od jednog sata i definirana je maksimalna količina podataka koja se u vremenskom intervalu zapisuje u MySQL bazu. Unutar pretplata, komponenta *statistika* definira koji su podaci potrebni za upis u MySQL bazu. Prilikom početka svakog novog vremenskog intervala definiraju se nove pretplate na sve podatke u svim područjima. Prikupljanjem podataka se te pretplate mijenjaju, tj. ukidaju se pretplate za područja s dovoljnim brojem prikupljenih mjerenja. Svaka poruka primljena od sustava *publish-subscribe* sadrži podatke o jednom mjerenju koji se zapisuju u za to predviđenu tablicu MySQL baze. Pretplata za statistiku je aktualna sve dok se od statistike pošalje poruka *unsubscribe* prema komponenti *broker*. Pretplate se ukidaju kad se prikupi dovoljno mjerenja ili objavljeni podaci pokriveni tom pretplatom postanu pre stari. Istekom vremenskog intervala (svakog punog sata) izračunavaju se srednje izmjerene vrijednosti za taj interval i takve izračunate vrijednosti se zapisuju u za to predviđenu tablicu MySQL baze. Srednje vrijednosti se izračunavaju na temelju *n* mjerenja za svaku pojedinu stavku koja se mjeri. Ako se za neko područje prikupi dovoljno mjerenja za izračun srednje vrijednosti prije isteka vremenskog intervala, ukida se pretplata za to područje i zapisuju se dobiveni podaci u MySQL bazu.

2.3.1 Idejno rješenje komponente statistika

Prilikom pokretanja aplikacije pokreću se konekcije prema komponenti *broker publish-subscribe* sustava i prema MySQL bazi podataka. Adrese i ostali podaci za ove konekcije su postavljene u priloženim datotekama unutar direktorija *configuration* ili se mogu eksplicitno zadati kroz grafičko sučelje.

Aplikacija provjerava dali je potrebno komponenti *broker* poslati podatke o područjima. Ako je ovaj uvjet zadovoljen, aplikacija se spaja na komponentu *broker* i objavljuje podatke o područjima.

Unutar svakog sata generiraju se i šalju *subscribe* poruke s novim pretplatama na informacije za svako pojedino područje. Komponenta *broker* je zadužena za obradu pretplata i filtriranje *publish* poruka koje šalju pametni telefoni.

Pristizanjem *publish* poruka s komponente *broker*, razvijena aplikacija *statistika* vrši sljedeće operacije:

- Pohranjuje podatke pojedinog mjerenja u MySQL bazu
- Provjerava za zadano područje dali je dobiveno dovoljno mjerenja jednog ili više zadanih parametara
- Ako je pristiglo dovoljno podataka:
 - Izračunava srednje vrijednosti na temelju pristiglih podataka
 - Zapisuje srednje vrijednosti u MySQL bazu
 - Uklanja staru pretplatu
 - Po potrebi šalje novu pretplatu u kojoj nisu navedeni parametri za koje je prikupljen dovoljan broj mjerenja.

Završetkom svakog vremenskog intervala izračunavaju se srednje vrijednosti za ona područja na kojima nije bilo obavljeno dovoljno podataka i dobivene vrijednosti se zapisuju u MySQL bazu.

Gašenjem aplikacije pokreće se postupak uklanjanja svih aktivnih pretplata na komponenti *broker* i raskida konekcija s komponentom *broker* i s MySQL bazom.

2.3.2 MySQL baza podataka

Naša baza podataka sastoji se od 3 tablice: *mjerenje*, *podrucje* i *statistikasat*.

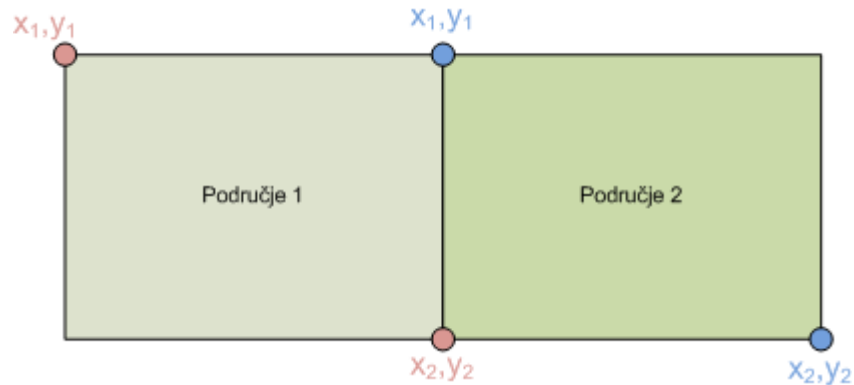
osjetizg.mjerenje	osjetizg.podrucje	osjetizg.statistikasat
idMjerenje : int(11)	idPodrucje : int(11)	idPodrucje : int(11)
vrijeme : datetime	latitude_1 : decimal(12,10)	datum : date
latitude : varchar(25)	longitude_1 : decimal(12,10)	vrijemePocetak : time
longitude : varchar(25)	latitude_2 : decimal(12,10)	vrijemeKraj : time
kolicinaCO : float	longitude_2 : decimal(12,10)	brMjerenja : int(11)
kolicinaNO2 : float	Naziv : varchar(50)	kolicinaCO : float
kolicinaO3 : float		kolicinaNO2 : float
kolicinaVOC : float		kolicinaO3 : float
temperatura : float		kolicinaVOC : float
vlaznost : float		temperatura : float
		vlaznost : float

Slika 9: ER model baze podataka

Opis tablica:

- *mjerenje* - Sadrži podatke o jednom mjerenju koji su iz sustava *publish-subscribe* prosljeđeni aplikaciji *statistika*. Za svako mjerenje bilježi se vrijeme, koordinate (x,y) lokacije mjerenja, te vrijednosti parametara mjerenja dobiveni očitavanjem senzora. U slučaju da jedan od parametara mjerenja nije izmjereno, na to mjesto se postavlja vrijednost „NULL“.
- *podrucje* - Sadrži podatke o područjima na koje grad podijeljen. Područja su kvadratnog oblika, a definirana su unaprijed. Svako područje jednoznačno je određeno dvjema točkama (x_1, y_1) , (x_2, y_2) .
- *statistikasat* - Agregatna tablica koja sadrži izračunatu srednju vrijednost mjerenja očitanih sa senzora za svako područja u definiranom vremenskom intervalu (trajanje pretplate, trenutno 1h). Pohranjuju se vrijednosti početaka i kraja intervala te srednje vrijednosti svih izmjerenih

parametara mjerenja. Atribut *brMjerenja* označava broj izvršenih mjerenja na temelju kojih je dobivena izračunata srednja vrijednost. Upisuje se vrijednost onog parametra mjerenja koji najmanje puta izmjeren. Ako se za neko područje u jednom vremenskom intervalu ne dobije nikakvo mjerenje, onda se za taj vremenski interval ne zapisuje redak u tablicu.



Slika 10: Određivanje granica područja

2.3.3 Poruke sustava Publish-Subscribe

Poruke koje se koriste za komunikaciju s *publish-subscribe* sustavom su tipa: *publish* (objava) i *subscribe* (pretplata). Aplikacija *statistika* koristi poruku *publish* za objavu podataka o svakom pojedinom području. U poruci se šalje identifikator područja, rubne koordinate područja i naziv. Podaci o područjima su potrebni za REST komunikaciju pametnih telefona s komponentom *broker*.

```
<?xml version="1.0" encoding='UTF-8'?>
<publication type="booleanHashtable">
  <validity>-1</validity>
  <attribute name="tipPoruke" operator="==" type="string">
    Podaci o podrucju
  </attribute>
  <attribute name='URLatitude' operator='==' type='double'>
    45.819947
  </attribute>
  <attribute name="URLongitude" operator="==" type="double">
    15.977497
  </attribute>
  <attribute name="LLlatitude" operator="==" type="double">
    45.819947
  </attribute>
  <attribute name="LLlongitude" operator="==" type="double">
    15.977497
  </attribute>
  <attribute name="Naziv" operator="==" type="string">
    Podsused
  </attribute>
  <attribute name="idPodrucje" operator="==" type="integer">
    1
  </attribute>
</publication>
```

Slika 11: Primjer *publish* datoteke s podacima jednog područja.

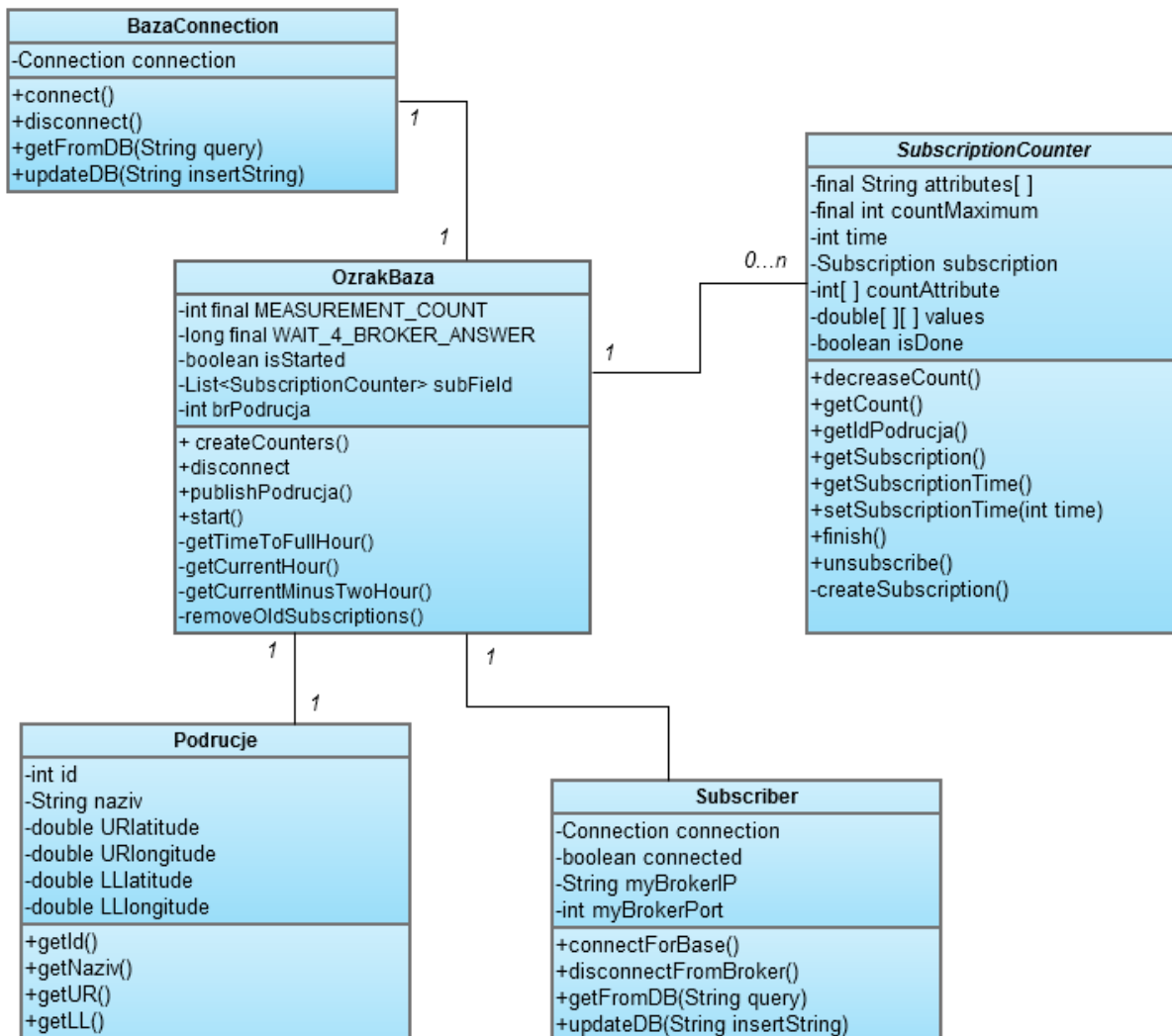
Poruke tipa *subscribe* se koriste za pretplaćivanje na potrebne informacije za pojedina područja. Ove poruke definiraju zahtjevano područje, vrijeme unutar kojega su sakupljeni podaci i tipove podataka koji su trenutno potrebni aplikaciji za upis u MySQL bazu. Ako aplikacija sakupi dovoljno podataka za određeni parametar, šalje se nova *subscribe* poruka u kojoj taj parametar nije naveden, a stara se uklanja s komponente broker slanjem *unsubscribe* poruke.

```
<?xml version="1.0" encoding='UTF-8'?>
<subscription type="booleanTriplet">
  <validity>-1</validity>
  <attribute name="tipPoruke" operator="==" type="string">Izmjereni
podaci</attribute>
  <attribute name="sat" operator="=" type="integer">10</attribute>
  <attribute name="dan" operator="=" type="integer">1</attribute>
  <attribute name="mjesec" operator="=" type="integer">1</attribute>
  <attribute name="godina" operator="=" type="integer">2010</attribute>
  <attribute name="podrucje" operator="=" type="integer">1</attribute>
  <!-- Ovi sljedeci atributi nisu obavezni -->
  <!-- Oni bi trebali naznacavati dali je baza zainteresirana za odredjeni
podatak -->
  <!-- Sada vise nije tip ovih podataka string nego double, kako bi se na
brokeru mogli usporedjivati -->
  <attribute name="kolicinaCO" operator=">" type="double">0</attribute>
  <attribute name="kolicinaNO2" operator=">" type="double">0</attribute>
  <attribute name="kolicinaO3" operator=">" type="double">0</attribute>
  <attribute name="kolicinaVOC" operator=">" type="double">0</attribute>
  <attribute name="temperatura" operator=">" type="double">0</attribute>
  <attribute name="vlaznost" operator=">" type="double">0</attribute>
</subscription>
```

Slika 12: Primjer *subscribe* datoteke s podacima pretplate

2.3.4 Implementacija sustava

Class Diagram



Slika 13: Dijagram klasa komponente statistika

Klasa `OzrakBaza` je centralna klasa u radu aplikacije. Prilikom pokretanja aplikacije uvijek se stvara samo jedna instanca ove klase. Stvaranjem nove instance ove klase, sustav se pokušava spojiti na MySQL bazu i na *broker* komponentu PUB-SUB sustava. Neuspjeh pri spajanju s ovim komponentama rezultira u bacanju iznimke i javlja pogrešku. Nakon uspješnog spajanja je moguće pokrenuti funkciju `start()` koja onda započinje s radom aplikacije.

Prvo se provjerava dali je uključena opcija za objavljivanje Područja u PUB-SUB sustav. Za slučaj kada je ta opcija uključena, poziva se metoda `publishPodrucja()` koja čita podatke iz MySQL baze i šalje obavijesti na komponentu *broker*.

Nakon inicijalne objave područja, dretva koja obrađuje klasu `OzrakBaza` ulazi u beskonačnu petlju. Prekidanjem rada ove dretve sustav se pokušava odspojiti od komponente *broker* i MySQL baze i aplikacija se zatvara.

Beskonačna petlja služi kao okidač za periodičke aktivnosti koje aplikacija izvršava, a to su:

- Periodičko stvaranje instanca klase `SubscriptionCounter` metodom `createCounters`.
- Pokretanje ukidanja starih pretplata iz PUB-SUB sustava.
- Provjera postojanosti podataka o područjima unutar PUB-SUB sustava (opcionalno).

Klasa `SubscriptionCounter` služi za evidenciju dolazećih podataka za jednu pretplatu i za ažuriranje te pretplate ukoliko je to potrebno. Također se u ovoj klasi izračunavaju srednje izmjerene vrijednosti i vrši se upis tih vrijednosti u MySQL bazu.

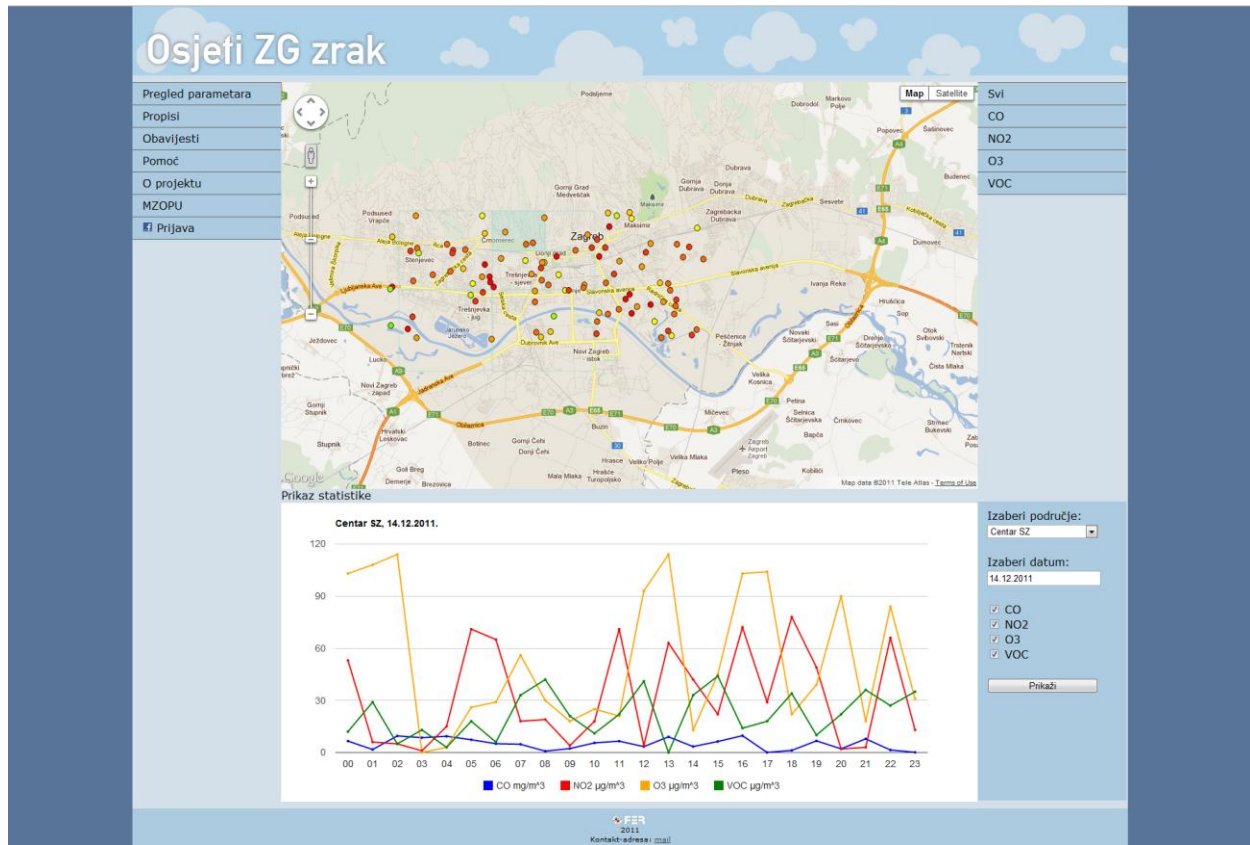
Metodom `createSubscription()` se stvara nova pretplata za područje koje je pridijeljeno instanci. Pretplata je definirana s područjem, vremenskim intervalom unutar kojeg je zadana. Valjanost svake pretplate je podešeno na 2 vremenska intervala.

Metodom `decreaseCount()` se naznačuje da je pristigao paket s podacima za područje za danu instancu klase. Ovo pokreće upis podataka u upis bazu podataka i po potrebi izračun i upis srednjih vrijednosti te poziv `finish()` metode.

Metoda `finish()` se pokreće u slučaju da je pretplati istekao rok valjanosti ili je prikupljen zadan broj mjerenja. Ovo pokreće uklanjanje pretplate s komponente *broker* i postupak izračunavanja srednjih vrijednosti.

2.4 Web aplikacija

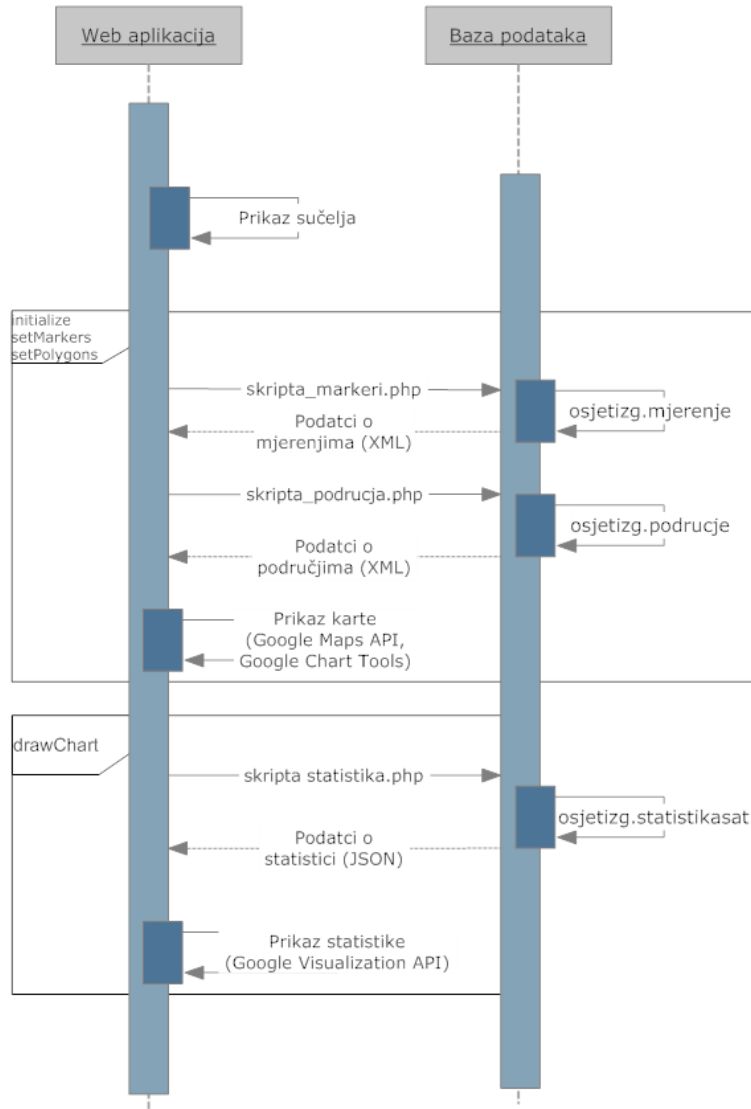
Web aplikacija (Slika 14) namijenjena je za prikaz podataka o kvaliteti zraka na mjenom području korisnicima. Korisnik može odabrati podatke o kvaliteti zraka za određeno vremensko razdoblje i područje od interesa. Podaci se nakon toga prikazuju na interaktivnoj karti podijeljenoj po područjima. Karta sadrži sva mjerenja zabilježena za taj vremenski period. Odabirom određenog područja prikazuje se statistika za odabrano područje u obliku grafa za svaki parametar mjerenja. Također, aplikacija omogućuje povezivanje korisnika sa Facebook društvenom mrežom. Ukoliko korisnik posjeduje profil na Facebooku, ima mogućnost podijeliti sa svojim prijateljima vrijednosti bilo kojeg mjerenja na bilo kojem području u grafičkom obliku. Vijest koju korisnik objavi pojaviti će se na njegovom zidu, tako da je i njegovi prijatelji mogu vidjeti.



Slika 14: Početni ekran web aplikacije

2.4.1 Korištenje podataka


Mjerni podaci korišteni za prikaz dohvaćaju se iz baze podataka na zahtjev korištenjem PHP skripti koje prema potrebi primaju parametre. Korištenje parametara omogućuje fleksibilnost u dohvat podataka, pa je tako moguće na zahtjev dohvatiti podatke za određeno vremensko razdoblje ili podatke o odabranim parametrima mjerenja koristeći istu skriptu. Podatci o mjerenjima se servisima Googlea iz tablice "mjerenja" u bazi podataka prenose u XML datotekama u kojima je osnovni element marker. Atributi osnovnog elementa sadrže podatke koje želimo prikazati poput koordinata, vremena, izmjerenih vrijednosti, boje ikone. Podaci o područjima iz tablice "podrucje" također se prenose u XML datoteci, dok se podatci za statistiku iz tablice "statistikasat" prenose u JSON formatu. Komunikacija između aplikacije i baze podataka prikazana je dijagramom na slici Slika 15.



Slika 15: Dijagram komunikacije između aplikacije i baze podataka

2.4.2 Traka izbornika

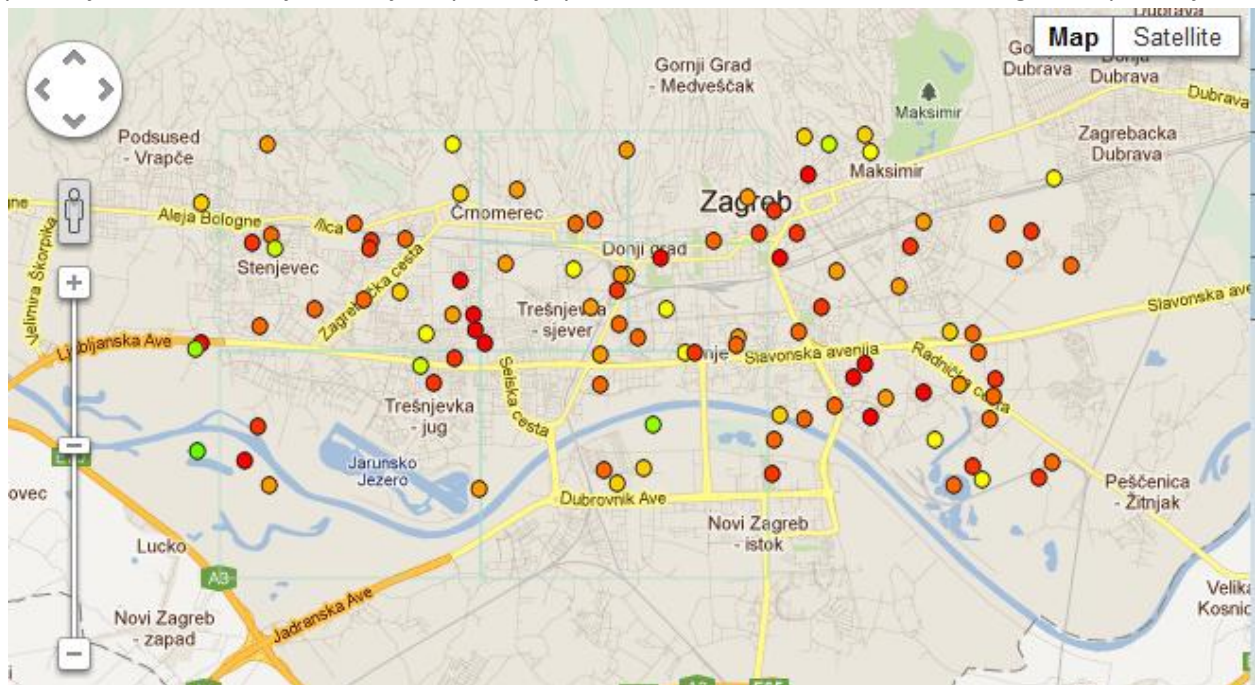
Traka izbornika sadrži nekoliko poveznica prema sadržajima vezanima uz prikaz i uz projekt. Vanjske poveznice "Propisi" i "MZOPU" vode prema stranicama Narodnih novina te Ministarstva zaštite okoliša, prostornog uređenja i građenja na kojima se mogu pronaći dodatne informacije o kvaliteti zraka, plinovima i okolišu. Preostale poveznice sadržaj prikazuju u Javascript prozoru unutar stranice i sadrže informacije poput parametrima na karti i uputa za korištenje. Izgled trake prikazan je na slici Slika 16.

Pregled parametara
Propisi
Obavijesti
Pomoć
O projektu
MZOPU
 Prijava

Slika 16: Traka izbornika

2.4.3 Interaktivna karta i prikaz parametara mjerenja

Za prikaz interaktivne karte korištene su usluge Google Maps API-ja. Google Maps API V3 je besplatna usluga dostupna svakoj nekomercijalnoj web stranici. Omogućuje ugradnju Google karte korištenjem JavaScripta te manipulaciju ugrađene karte i sadržaja koji se na njoj prikazuje. Karta je podijeljena na područja koja su animirana, pa tako kada korisnik prelazi mišem preko područja ispisuje se naziv tog područja, a kvadrant koji omeđuje to područje potamni kako bi se vizualno istaknule granice područja.



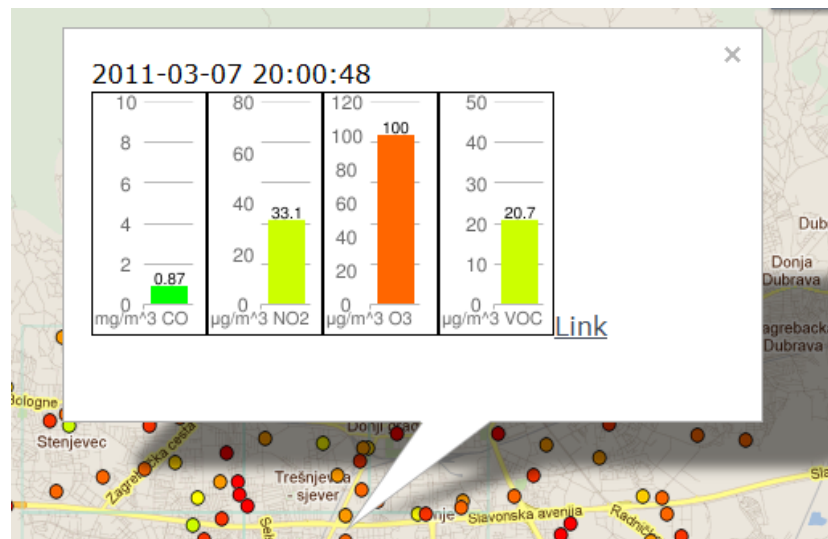
Slika 17: Interaktivna karte dobivena korištenjem Google Maps API-ja

Podaci o kvaliteti zraka prikazuju se u obliku točaka - markera - na karti. Svaki marker predstavlja izmjerene podatke o kvaliteti zraka na određenom mjestu. Boje markera (Slika 18) vizualno prikazuju izmjerenu kvalitetu zraka i intuitivne su korisnicima, pa tako crveni marker označava da je izmjerena vrijednost bilo kojeg parametra na tom mjestu na kritičnoj razini, dok zeleni marker označava da su izmjerene vrijednosti niske, tj. da je zrak dobre kvalitete.



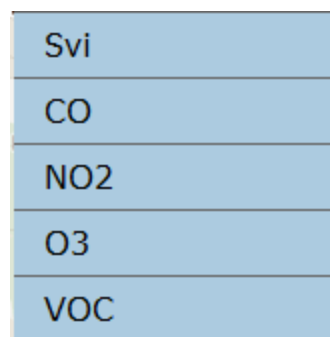
Slika 18: Markeri za označavanje izmjerenih vrijednosti

Prelaskom miša preko bilo kojeg markera otvara se skočni prozor u kojem su prikazani detaljni podaci parametara kvalitete zraka koji su zabilježeni za odabrano mjerenje (Slika 19). Na slici možemo vidjeti točan datum i vrijeme kada je mjerenje očitano na području, kao i grafički prikaz s očitanim vrijednostima svakog parametra mjerenja. Sve izmjerene vrijednosti parametra izražene su u standardnim mjernim jedinicama koje su propisane od državnog hidrometeoroškog zavoda. Grafički prikaz svake vrijednosti također vizualno odražava razinu očitane vrijednosti unutar definiranih kritičnih vrijednosti. Boje korištene za vizualizaciju podatka iste su kao kod vizualizacije markera, te stoga omogućuju konzistentan pregled i lakše razumijevanje značenja podataka. Sadržajem prozora također se upravlja putem Google Maps API-ja, dok su podaci prikazani u grafičkom obliku dobiveni korištenjem Google Chart Tools-a.



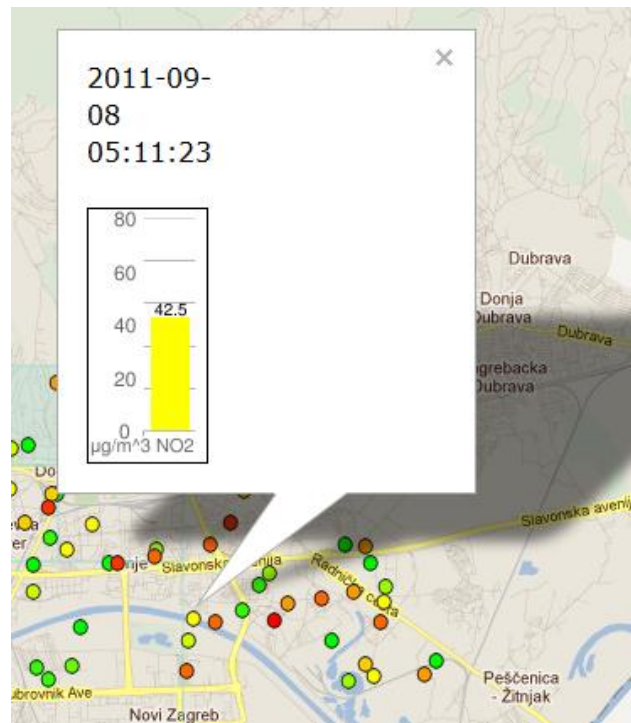
Slika 19: Skočni prozor s detaljnim podacima odabranog mjerenja

Uz pretpostavljeni prikaz svih parametara mjerenja korisnik može i filtrirati prikaz tako da odabere jedan od ponuđenih parametara po kojem želi analizirati mjerenja. Odabir parametra po kojem će se filtrirati prikaz obavlja se klikom na ponuđene vrijednosti u desnom dijelu početnog ekrana web aplikacije (Slika 20).



Slika 20: Odabir željenog parametra mjerenja

Klikom na NO₂ mjerenja se filtriraju i prikazuje se samo NO₂ parametar mjerenja (Slika 21). Boje markera se također mijenjaju prema razinama vrijednostima koje su izmjerene. Promjena markera na karti se odvija bez ponovnog učitavanja cijele stranice, a izvedena je korištenjem JavaScripta kojim se poziva Google Maps API s novim vrijednostima parametara.



Slika 21: Prikaz mjerenja za parametar NO_2

2.4.4 Prikaz statistike za odabrano područje

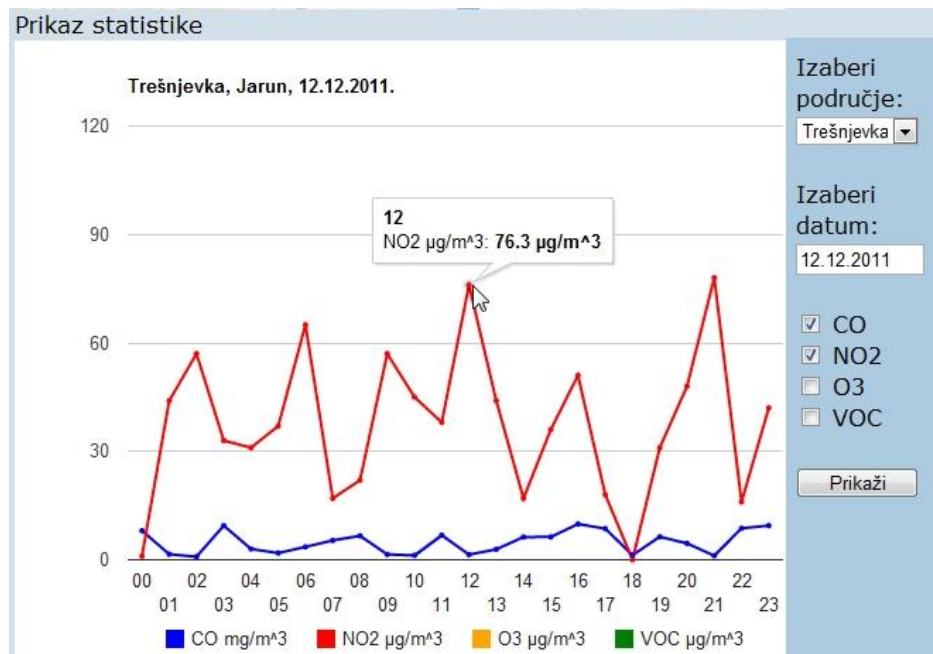
Statistika se prikazuje u obliku grafa za odabrano područje. Na grafu su srednje vrijednosti mjerenja kroz sate u jednom danu. X-os predstavlja vrijeme od 0-24h, a na y-osi su izračunate vrijednosti dane u mg/m^3 ili $\mu\text{g}/\text{m}^3$. Korisnik može odabrati područje i datum za prikaz statistike te koji će se plinovi prikazati na grafu.

Datum se izabire klikom na polje za unos nakon čega se otvara prozor s kalendarom (Slika 22). Ova je funkcionalnost izvedena pomoću jQuery dodatka DatePicker.



Slika 22: Odabir datuma za prikaz statistike

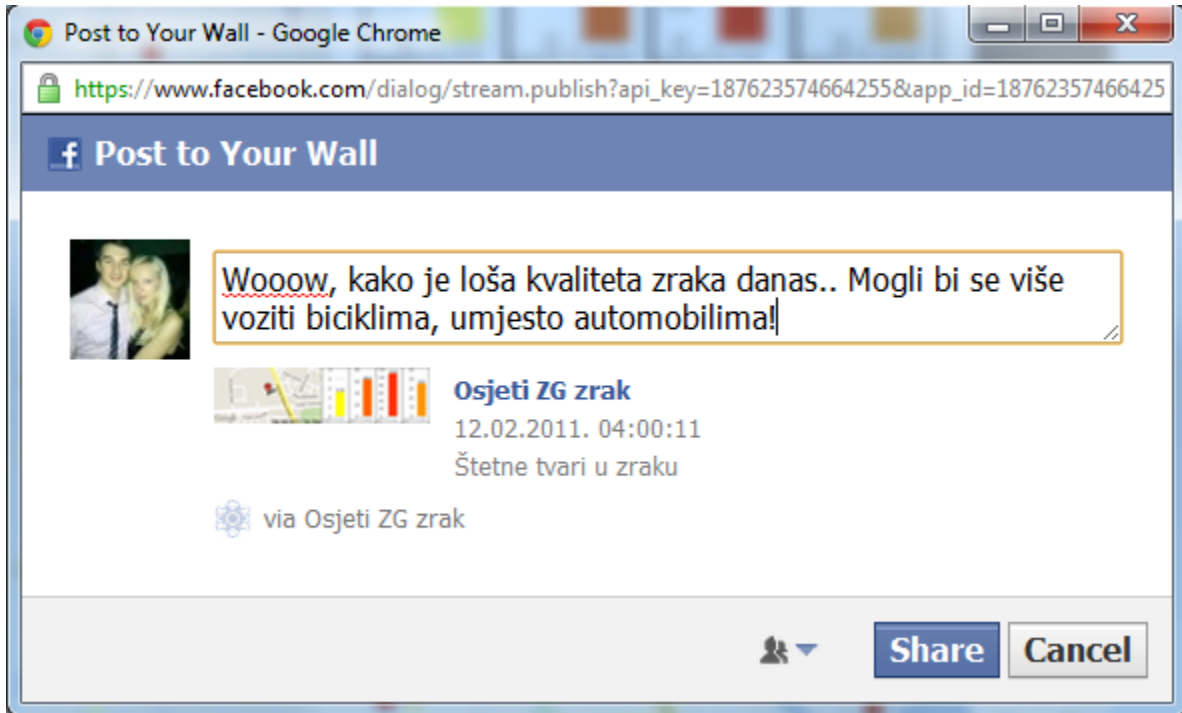
Područje se može izabrati iz padajućeg izbornika s popisom svih područja ili klikom na područje na karti što automatski označava područje i u izborniku. Izbor plinova koji će biti prikazani u statistici vrši se odabirom checkboxa za svaki plin. Nakon odabira svih parametra, klikom na gumb "Prikaži" poziva se funkcija za crtanje koja iscrtava graf. Ovo je ostvareno pomoću Google Visualization API-ja. Google Visualization API pruža jednostavan način povezivanja grafova s izvorom podataka putem Javascripta i ugradnju u web stranice. Također nudi i velike mogućnosti prilagodbe. Graf je interaktivan i prelaskom miša preko pojedine točke prikazuje se točna izračunata vrijednost (Slika 23).



Slika 23: Izgled grafa statistike za zadane parametre (prikazani samo CO i NO2)

2.4.5 Objavljivanje sadržaja na društvenoj mreži Facebook

Korisnik može bilo koje mjerenje prikazano skočnim prozorom objaviti na Facebooku i tako obavijestiti svoje prijatelje o kvaliteti zraka na nekom području. Preduvjet za objavu na Facebooku jest da je korisnik vlasnik Facebook profila. Vijest se objavljuje korištenjem Facebook *feed plugin*a za internetske stranice. *feed plugin*om je moguće objaviti bogati sadržaj na zidu korisnika, poput slike, poveznice, audio i video zapisa te teksta. Implementirani *feed plugin* omogućuje objavu grafičkog zapisa mjerenja u obliku slike, sa tekstualnom porukom koja opisuje vrijednosti izmjerenih parametara te poveznicom na službenu stranicu Osjeti ZG zrak projekta.



Slika 24: Facebook feed dijalog kojim se vijest objavljuje na Facebook profilu korisnika

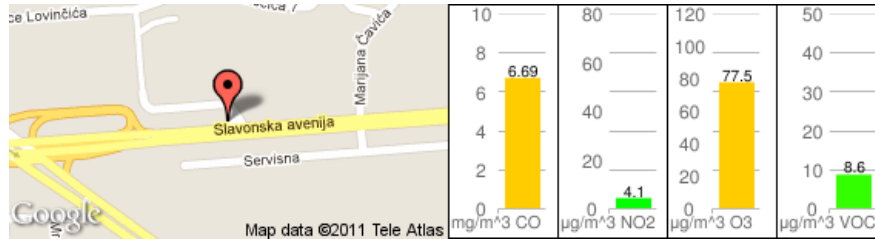
Nakon što korisnik objavi vijest putem web stranice na svom Facebook profilu, vijest se vidi na korisnikovom profilu, njegovi prijatelji je mogu komentirati i dalje dijeliti.



Slika 25: Uspješna objava na Facebook profilu korisnika.

Kako bismo podijelili sliku na Facebook profilu korisnika potrebno je u feed dijalog umetnuti poveznicu na tu sliku što znači da se ona mora nalaziti negdje na webu. Kako se slike za pojedino mjerenje dinamički generiraju pozivanjem Google Chart Tools-a koji ih vraća u obliku HTML `` taga bilo je potrebno osmisliti način kako tim slikama dodijeliti web adresu. Taj posao obavlja PHP skripta

"skripta_merge.php" kojoj se predaju parametri potrebni za generiranje slike. Skripta generira URL-ove prema Google Chart Toolsu i Google Maps API-ju i tako dohvaća slike grafova i sliku dijela karte na kojem se nalazi lokacija mjerenja. Te slike zatim spaja u jednu i vraća njenu adresu kako bi se mogla objaviti na Facebooku (Slika 26).



Slika 26: Dinamički generirana slika s kartom i grafovima

Literatura

- [1] Java SE 6 documentation, Oracle, <http://docs.oracle.com/javase/6/docs/>, 14. prosinca 2011.
- [2] Android The Developer's Guide, Google, <http://developer.android.com/guide/>, 14. prosinca 2011.
- [3] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, W3C, <http://www.w3.org/TR/CSS21/>, 14. prosinca 2011.
- [4] jQuery JavaScript framework, John Resig and the jQuery Team, <http://jquery.com/>, 14. prosinca 2011.
- [5] jQuery UI, Paul Bakaus and the jQuery UI Team, <http://jqueryui.com/>, 14. prosinca 2011.
- [6] MySQL Documentation, MySQL AB, <http://dev.mysql.com/doc/>, 14. prosinca 2011.
- [7] PHP Documentation, The PHP Group, <http://www.php.net/docs.php>, 14. prosinca 2011.
- [8] Apache HTTP Server Version 2.2 Documentation, The Apache Software Foundation, <http://httpd.apache.org/docs/2.2/>, 14. prosinca 2011.
- [9] EZ430-RF2560 wiki, Texas Instruments, <http://processors.wiki.ti.com/index.php/EZ430-RF2560>, 14. prosinca 2011.
- [10] MSP430BT5190 product information, Texas Instruments, <http://www.ti.com/product/msp430bt5190>, 14. prosinca 2011.
- [11] Gas sensors data sheets and application notes, e2v, <http://www.e2v.com/products-and-services/instrumentation-solutions/gas-sensors/>, 14. prosinca 2011.

Suglasan s dokumentom (potpisuju članovi tima):

Josip Biščan

Datum: _____ Potpis: _____

Filip Gvardijan

Datum: _____ Potpis: _____

Hrvoje Ilić

Datum: _____ Potpis: _____

Daniel Petrač

Datum: _____ Potpis: _____

Ruben Štulić

Datum: _____ Potpis: _____

Lucija Zadrija

Datum: _____ Potpis: _____

Odobrio(potpisuje nastavnik):

Vedran Bilas

Datum: _____

Potpis: _____

Maja Matijašević

Datum: _____

Potpis: _____

Vedran Podobnik

Datum: _____

Potpis: _____

Ivana Podnar Žarko

Datum: _____

Potpis: _____