

Ivan Pašić, univ. bacc. ing. geod. et geoinf. ▶ diplomski studij, Geodetski fakultet, Sveučilište u Zagrebu, Kačićeva 26, 10000 Zagreb, e-mail: ivpasic@geof.hr



TagIT – Sustav za satelitsko praćenje objekata u realnom vremenu

SAŽETAK: Projekt predstavljen u ovom radu nastao je u sklopu natjecanja "App Start Contest" 2014. godine koje je organizirala studentska udruga eStudent. Projekt se na natjecanju predstavlja pod nazivom TagIT. Cilj projekta bio je izraditi internetsku GIS aplikaciju za satelitsko praćenje različitih objekata u realnom vremenu te uz to, izraditi i vlastiti GPS uređaj pomoću kojeg bi se demonstrirao rad samog sustava. Također, pri samom razvoju sustava, pazilo se da sve komponente sustava budu izrađene koristeći isključivo tehnologije otvorenog koda (eng. open source). Stoga je i sam GPS uređaj izrađen pomoću popularne platforme otvorenog koda Arduino. U ovom radu opisan je princip rada sustava te je dan pregled tehnologija korištenih za izradu sustava. Također su prikazane i neke od mogućnosti te funkcija same internetske aplikacije.

KLUČNE RIJEČI: satelitsko praćenje, GIS, Arduino, App Start Contest, TagIT

TagIT – Real-Time Tracking System

ABSTRACT: The project presented in this article is result of App Start Contest 2014. which was organized by the student association eStudent. The project participates on contest under name TagIT. The main idea of the project was to develop real-time web GIS application for satellite tracking of different objects and beside that, to personally build GPS device which would be used to demonstrate how system works. Beside that, while developing the system, one of the request was to use only open-source technologies. Following that idea, even the GPS device was built using popular open-source platform called Arduino. In this article it's described how system works and what technologies were used to develop it. There are also presented some of web app features and functionalities.

KEYWORDS: real-time tracking, GIS, Arduino, App Start Contest, TagIT

1. UVOD

U današnjem društvu spoznaja o lokaciji pojedinog objekta te o njegovoj brzini i smjeru kretanja postaje sve važnija. Tako smo svjedoci sve veće upotrebe različitih sustava za praćenje objekata u realnom vremenu koji se najčešće baziraju baš na upotrebi GPS-a. Ubrzanim razvojem tehnologije, kako hardvera tako i softvera, otvaraju se mogućnosti za dodatnim razvojem takvih sustava.

Većina takvih rješenja danas je i dalje iznimno skupa te se njihovo korištenje najčešće svodi na upotrebu u sustavima za upravljanje voznim parkom (eng. *fleet management*) i srodnim granama.

Kroz ovaj projekt pokušalo se, koristeći najnovije tehnologije, izraditi sustav koji bi bio kvalitetan i pouzdan, a u isto vrijeme i jednostavan za upotrebu, te tako široko primjenjiv.

2. ZAHTJEVI SUSTAVA

Pri izradi ovog projekta zadano je više zahtjeva koje bi sustav trebao zadovoljiti. Prije svega, sustav treba omogućiti pouzdan i točan prikaz položaja i kretanja GPS uređaja u realnom vremenu. Ipak, treba naglasiti kako, unatoč tome što se ovdje govori o prikazu u realnom vremenu, kod internetskih GIS aplikacija tako nešto zapravo ne postoji. Naime, od trenutka kada GPS uređaj dobije satelitski signal pa do trenutka kada se ti podaci obrade i pošalju na server te prikažu na ekranu korisnika prođe određeno vrijeme. Dakle, to teoretski više nije prikaz u realnom vremenu. Međutim, taj vremenski interval je vrlo kratak (često se mjeri u milisekundama) pa ćemo se u nastavku ipak služiti pojmom 'realno vrijeme'.

Osim spomenutog zahtjeva, sama mrežna GIS aplikacija bi uz informacije o kretanju trebala sadržavati i dodatne mogućnosti koje korisniku olakšavaju praćenje uređaja. Stoga je unutar nje omogućeno i definiranje područja unutar kojeg bi se GPS uređaj trebao kretati (eng. *geofencing*). Ako GPS uređaj izađe izvan tog područja, vlasnik uređaja tada na svoju e-adresu dobiva obavijest s podacima o tom događaju. Ta poruka sadrži podatke o trenutku napuštanja područja s točnim koordinatama uređaja u trenutku napuštanja.

Nadalje, mrežna aplikacija, uz pregled trenutne pozicije uređaja, mora korisniku omogućiti i pregledavanje povijesti kretanja te neke statističke pokazatelje (trajanje puta, vrijeme provedeno unutar ili izvan dozvoljenog područja, itd.).

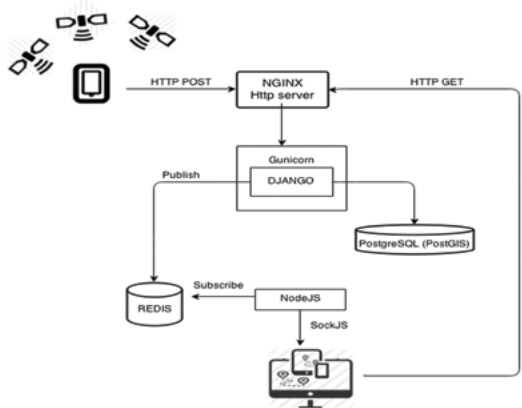
Jedan od zahtjeva sustava bio je i da mrežno sučelje bude prilagođeno ekranu uređaja na kojem se pregledava. Na taj je način omogućena jednaka preglednost aplikacije na mobilnim telefonima, tablet-uređajima te osobnim računalima.

3. ARHITEKTURA I DIZAJN SUSTAVA

Glavne sastavnice sustava su :

- GPS uređaj
- Server (poslužitelj)
- Baza podataka
- Klijent (web browser).

Odnos pojedinih dijelova sustava prikazan je na slici 3.1.



Slika 3.1. Prikaz arhitekture sustava

GPS uređaj (koji je povezan s razvijenom aplikacijom) samostalno je izrađen pomoću Arduino platforme. Arduino je platforma za izradu različitih elektronskih uređaja koji putem različitih senzora ostvaruju interakciju sa svojom okolinom. Glavna sastavnica svake Arduino ploče je mikrokontroler, a programski jezik koji se koristi za programiranje Arduino uređaja baziran je na C++ programskom jeziku (URL-1). U ovom projektu korištena je Arduino Uno ploča.

Specifikacije uređaja

Izrađeni GPS uređaj sastoji se od 3 glavne sastavnice. To su:

- Arduino Uno (mikrokontroler).
- GPRS modul (dodatni modul koji omogućuje korištenje SIM kartice mobilnog operatera tj. omogućuje uspostavu internetske veze te slanje HTTP zahtjeva na server).
- GPS modul + eksterna antena (dodatni modul koji pomoću satelitskog signala omogućuje određivanje položaja i dodatnih parametra vezanih uz kretanje uređaja (brzina, nadmorska visina, kurs...)).

Osnovni izgled izrađenog GPS uređaja prikazan je na slici 3. 2.



Slika 3.2. Izrađeni GPS uređaj

Princip rada izrađenog uređaja prilično je jednostavan. Putem vanjske antene koja je priključena na uređaj primaju se satelitski signali te se na temelju njih određuje trenutni položaj uređaja. Zatim se pomoću GPRS modula i SIM kartice mobilnog operatera uspostavlja internetska veza te se primljeni podaci (uz prethodnu obradu) šalju na server putem HTTP POST zahtjeva. Uređaj je programiran na način da konstantno određuje svoj položaj i ostale parametre, međutim, određeno je da te podatke šalje na server tek svakih 5 sekundi. Taj interval se lako može promijeniti ovisno o namjeni uređaja. Serverski dio aplikacije izrađen je uz upotrebu različitih tehnologija čiji je kratki pregled dan u nastavku teksta.

3.1. OPIS KORIŠTENIH TEHNOLOGIJA

Većina serverskog dijela aplikacije izrađena je pomoću Django. Django je mrežno razvojno okruženje otvorenog koda napisano u programskom jeziku Python. To je framework visoke razine koji potiče brz razvoj i čist pragmatičan dizajn (URL-2). U ovom projektu Django je korišten u prvom redu za izradu REST sučelja internetske aplikacije.

Uz Django, vrlo bitna tehnologija koja se koristila je i Redis. Redis se često definira kao sustav za pohranjivanje podataka u obliku 'ključ vrijednost' (eng. *key-value*), a glavna mu je prednost njegova iznimna brzina.

U ovom projektu Redis je korišten za implementaciju različitih funkcionalnosti. Tako se primjerice podaci (koordinate) koje šalje GPS uređaj spremaju u Redis listu, na temelju koje se kasnije, po završetku sesije, stvara pojednostavljena polilinja.

Ta pojednostavljena linija tada se sprema u PostgreSQL bazu podataka te predstavlja kretanje tog GPS uređaja za tu sesiju tj. vremenski interval.

Pritom treba naglasiti da se za stvaranje te polilnije ne koriste svi podaci iz Redis liste već samo dio podataka (npr. svaka treća koordinata), ovisno o broju podataka u listi. Naime, ako je uređaj bio aktivan duže vrijeme, tada i sama lista sadrži više podataka (koordinata) te ne bi imalo smisla kreirati polilinjiju koja bi obuhvaćala sve koordinate. Dovoljno je kretanje uređaja prikazati pojednostavljenom polilinjijom koja obuhvaća samo dio koordinata.

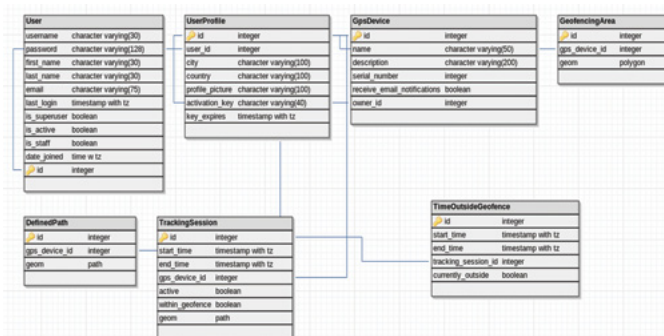
Osim spomenutog primjera, Redis je u ovom projektu također poslužio i kao *messaging system*. Naime, Redis omogućuje jednostavnu provedbu *publish/subscribe* metode koja je jedan od ključnih principa na kojima se baziraju *real-time web* GIS aplikacije.

Uz Django i Redis, za serverski dio aplikacije u manjem opsegu korišten je i NodeJS. NodeJS je programska platforma bazirana na JavaScript programskom jeziku koja služi za jednostavno izradu brzih, skalabilnih mrežnih aplikacija (URL-3).

Naime, Django nije primarno namijenjen izradi mrežnih aplikacija u realnom vremenu pa je bilo potrebno na drugi način razviti sustav gdje je stalno otvorena veza između klijenta i poslužitelja tj. gdje je potrebno slanje obavijesti sa servera na klijenta. Dakle, NodeJS je u ovom projektu poslužio samo kao jednostavni *push* servis na način da čeka poruke na Redis kanalu (*subscribe*) te u slučaju poruke šalje obavijesti klijentu (pregledniku).

Kao što je već spomenuto, za izradu mrežnih aplikacija u realnom vremenu potrebna je dvosmjerna komunikacija između klijenta i poslužitelja. Dolaskom HTML5 tehnologije ta je komunikacija omogućena korištenjem Websocketeta. Ipak, Websocketeti su nova tehnologija te ih još uvijek ne podržavaju svi preglednici. Stoga se u ovom projektu koristi i JavaScript biblioteka SockJS koja zapravo obuhvaća WebSocket tehnologiju odnosno koristi sve mogućnosti Websocketeta kada ih klijent podržava, ali u slučaju da klijent ne podržava Websocketete, SockJS i dalje omogućava uspostavu dvosmjerne komunikacije.

Za pohranu podataka o registriranim korisnicima i GPS uređajima te njihovim postavkama (*geofencing* područje, linija kretanja) korištena je PostgreSQL baza podataka sa svojim prostornim proširenjem PostGIS. Model baze podataka prikazan je na slici 3.1.1.



Slika 3.1.1. Model baze podataka

Treba naglasiti kako se podaci GPS uređaja koji stižu na server ne pohranjuju direktno u PostgreSQL bazu podataka već se za taj dio koristi već opisani princip s Redis-om.

Naime, s obzirom na to da se ovdje radi o velikom broju podataka koji stižu na server to bi bio izuzetno velik broj zapisa pa njihovo pohranjivanje u relacijsku bazu podataka ne bi imalo previše smisla. Osim toga, takav način upravljanja podacima previše bi opteretio sustav te bi se uvelike otežalo održavanje samog sustava.

U konačnici, za izradu samog mrežnog sučelja aplikacije korištene su standardne *front-end* tehnologije (HTML5, CSS3, JavaScript) uz upotrebu dodatnih biblioteka. Tako je primjerice za izradu same interaktivne karte korišten LeafletJS (biblioteka otvorenog koda napisana u JavaScript programskom jeziku, a namijenjena upravo izradi interaktivnih mrežnih karti) dok je za dizajn cjelokupnog sučelja korišten Twitter Bootstrap koji je s vremenom postao nezaobilazan izbor kada je riječ o izradi jednostavnih i elegantnih sučelja prilagođenih uređajima svih veličina (mobilnim telefonima, *tablet*-uređajima i računalima).

4. NAČIN KORIŠTENJA SUSTAVA

Cilj ovog projekta bio je omogućiti jednostavno korištenje sustava te se stoga pokušalo cijeli postupak učiniti što jednostavnijim za upotrebu.

Svaki korisnik (vlasnik GPS uređaja) koji želi pratiti kretanje svog uređaja mora prvo napraviti korisnički račun na internetskoj stranici <http://www.tagit-easy.com>.

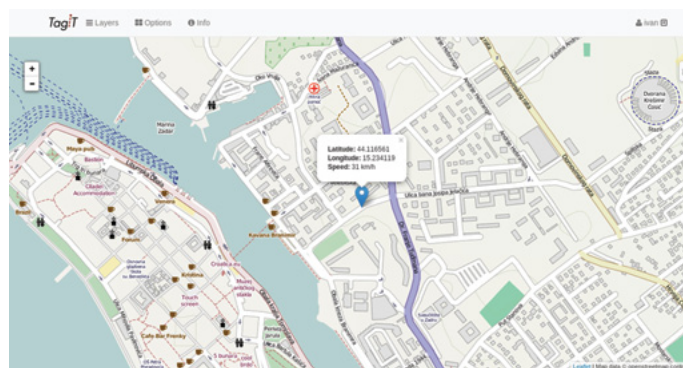
Nakon što napravi račun može se registrirati u sustav gdje mu se nudi mogućnost registriranja GPS uređaja kojeg želi pratiti. Za registraciju uređaja potrebno je unijeti podatak o serijskom broju uređaja (to je podatak jedinstven za svaki GPS uređaj, a dobiven je uz sam uređaj) te, po želji, neke dodatne atribute. Uz to vlasnik uređaja definira i želi li primati obavijesti na svoju e-adresu, u slučaju da uređaj napusti dozvoljeno područje.

Nakon što korisnik uspješno registrira GPS uređaj, pruža mu se mogućnost definiranja postavki vezanih za samo praćenje uređaja. Tako primjerice ima opciju definiranja (označavanja na karti) područja unutar kojeg želi da se njegov uređaj kreće (slika 4.1.) te definiranje linije, tj. putanje kojom se uređaj treba kretati.



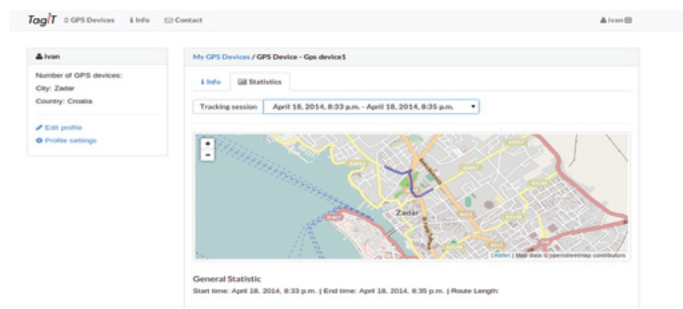
Slika 4.1. Zadavanje dopuštenog područja kretanja GPS uređaja

Nakon što je korisnik uspješno registrirao svoj GPS uređaj može započeti s njegovim praćenjem na način da putem mrežnog sučelja pristupi interaktivnoj karti na kojoj se u realnom vremenu prikazuje trenutni položaj GPS uređaja (slika 4.2.).



Slika 4.2. Prikaz interaktivne karte za praćenje GPS uređaja

Uz interaktivnu kartu, korisnik unutar nadzorne ploče ima pristup i drugim podacima vezanim za kretanje tog GPS uređaja (slika 4.3.). Na taj je način omogućen prikaz povijesti kretanja za pojedinu sesiju kao i prikaz statističkih podataka o pojedinoj sesiji (početak praćenja, završetak praćenja, duljina prijeđenog puta, vrijeme provedeno unutar ili izvan dopuštenog područja).



Slika 4.3. Prikaz povijesti kretanja GPS uređaja

5. ZAKLJUČAK

Izuzetno brz razvoj tehnologije otvara neke nove mogućnosti pri razvoju mrežnih GIS aplikacija u realnom vremenu.

Projekt "TagIT" koji je predstavljen u ovom radu samo je jedan primjer takvih aplikacija, no dovoljan je da pokaže kako se upotrebom novih tehnologija (i to onih otvorenog koda) može razviti kompletan sustav.

S obzirom na to da je sustav još u razvoju te da postoji još dosta ideja koje se planiraju provesti, nezahvalno je donositi zaključke o funkcionalnosti samog sustava. Ipak, nakon svega što je do sada učinjeno, može se reći kako su sva prvotna očekivanja sustava ispunjena te kako je sustav već u ovom obliku spreman za korištenje.

Iako razvoj ovakvih aplikacija nije primarna geodetska djelatnost, bilo bi odlično kada bi ovakav i slični projekti potaknuli geodetsku, ali i geoinformatičku struku da prošire svoje djelatnosti. Tehnologija svakodnevno ide naprijed te se mogućnosti razvoja ovakvih sustava iz dana u dan povećavaju. Pritom se ne misli samo na razvoj aplikacija, već i na razvoj samih GPS uređaja. Naime, upotrebom različitih integriranih senzora (primjerice žiroskopa i akcelerometra) može se dodatno povećati točnost GPS uređaja, te omogućiti pozicioniranje čak i unutar zatvorenih prostora. Upravo je to jedan od daljnjih planova u razvoju TagIT sustava.

LITERATURA

- › URL-1: <http://arduino.cc/en/Guide/Introduction> (27. 04. 2014.)
- › URL-2: <http://www.djangoproject.com/> (27. 04. 2014.)
- › URL-3: <http://nodejs.org/> (27. 04. 2014.)