SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1917

WEB APLIKACIJA ZA PRIKAZ I UNOS GEOGRAFSKIH PODATAKA PLANINARA

Anto Miketa

Zagreb, rujan 2012.

Tekst zadatka s potpisima

Sadržaj

Zahvala

Hvala mojoj ženi na strpljenju i potpori koju mi je uvijek pružala. Za sve vrijeme koje sam proveo radeći umjesto provodeći ga s njom i svu ljubav koju mi pruža svaki dan.

Hvala mojim kćerima jer su razumjele da tata radi zadaću i zagrlile me kad sam se u pauzi igrao s njima.

Hvala mojim roditeljima koji su me odgojili i othranili. Hvala za svu ljubav i strpljenje koji su imali tjekom godina. Hvala im što sam uz njih naučio što je stvarno važno u životu i što mi i sada pomažu to dostići.

Hvala braći i sestri jer su mi bili potpora i kritičari.

Hvala obitelji Dukić jer me je prihvatila i imala razumjevanja.

Hvala mojoj mentorici Prof. dr. sc. Željki Mihajlović na znanju koje je prenijela na mene i pomoći koju mi je pružila u izradi ovog rada. Hvala na otvorenu i brzu komunikaciju, motiviranost i povjerenje koje ste mi iskazali. Radeće ovaj rad ponovno sam zavolio programiranje.

Hvala svima koji su molili za mene jer sam zahvaljujući njihovim molitvama uspio završiti ovaj studij.

UVOD

Već godinama sam rekreativni planinar i iako postoji mnoštvo knjiga o planinarstvu u Hrvatskoj na internetu je teško pronaći točne informacije na jednom mjestu. Zato sam se odlučio za izradu ove aplikacije kako bi na jednom mjestu pružio mogućnost prikaza svih planinarskih objekata u Hrvatskoj (po potrebi i šire).

Inspiracija su mi bile knjige „Hrvatske planine“ dr. Željka Poljak i „Planinarski vodič po Hrvatskoj“ Alana Čaplar. U njima je moguće pronaći informacije o gotovo svim planinarskim objektima i znamenitostima u Hrvatskoj.

Aplikacija je razvijena na google maps API-ju jer je isti dobro prihvaćen i jako raširen te besplatan. Radi se o web aplikaciji, a izabrane su sljedeće tehnologije za razvoj. Korišten je html za kao osnova i jezik za prikaz elemenata na ekranu. Na klijentskoj strani korišten je javascript te jedna od najpoznatijih biblioteka za javascript jquery sa svojim grafičkim sučeljem. Za serverski jezik odabran je php, a aplikacija se izvodi na apache web serveru, konkretno radi se o xampp-u čiji je sastavni dio i mysql koji je korišten za bazu podataka. Ove tehnologije izabrane su jer su otvorenog koda (engl. open source) i besplatne te je moguće naći mnogo literature i pomoći na internetu.

Aplikacija je podijeljena na nekoliko cjelina. U prvoj se obrađuju planinarske lokacija (planinarski domovi, skloništa i vrhovi). Osnovni google maps marker je crvene boje s točkom u sredini, ali kako bi objekte bilo lakše raspoznati svaka vrsta objekta prikazuje se drugačijim markerom. Prilikom ulaska u aplikaciju prikazuju se svi objekti, a naknadno je moguće prikazati specifične objekte filtriranjem po nadmorskoj visini i vrsti objekta.

Nadalje moguće je prikazati planinarske staze. Staze se iscrtavaju na kartu, a moguće je i dobiti malo detaljnije informacije o pojedinoj stazi.

U sljedećoj cjelini prikazuju se geografske informacije o planinarima. Moguće je prikazati trenutne lokacije planinara i putanje po kojima su pojedini planinari išli u određenom vremenskom razdoblje.

Zadnji dio aplikacije je mjerenje. Moguće je mjeriti udaljenosti i površine. U izračun je uračunata zakrivljenost zemlje.

Aplikacija je dovršena, ali također ima mnogo mogućnosti za unapređenje i daljnji razvoj. Dva su glavna smjera u kojima bi se aplikacija mogla dalje razvijati. Prvi je web aplikacija otvorenog tipa (dozvoljen pristup bez autorizacije). U tom slučaju daljnji razvoj temeljio bi se na prikazu i proračunavanju dodatnih podataka o lokacijama i stazama. Za staze bi se mogao prikazati i reljefni presjeci te bi se mogla računati duljina staze. Ako bi aplikacija išla u drugom smjeru poradili bismo na sustavu za praćenje planinara (ili bilo kojih drugih subjekata). Trebalo bi napraviti sustav za upisivanje lokacija u bazu (stvarni podatci), npr. aplikacija za mobitele koja bi te podatke slala. Također u ovom slučaju aplikacija ne bi bila dostupna svima pa bi bilo potrebno napraviti sustav za autorizaciju i administraciju prava korisnika. Tema rada nije bila sigurnost, ali i na tom području su moguća unaprjeđenja u nastavku razvoja. Prvenstveno obrana od najjednostavnijih napada kao SQL injection.

Korištene su tehnologije otvorenog koda kako bi projekt bio što dostupniji, jeftiniji i lakši za buduću nadogradnju. Iz tog razloga su birane tehnologije preporučene LAMP filozofijom.

Tehnologije

HTML

Hyper text Markup language je jezik za opis web stranica. Html nije programski jezik i koristi oznake (eng. tags) za opis. Oznake uglavnom dolaze u paru (npr.) i jedna označava početak, a druga kraj i između njih se obično dodaje tekst.

Prvi dokument koji opisuje HTML izdan je 1990. I zvao se „HTML Tags“, a autor je bio Tim Berners-Lee. Od tada se HTML stalno razvija. Od 1996. HTML specifikacije održava World Wide Web Consortium (W3C) iako je 2000. HTML postao međunarodni standard (ISO/IEC 15445:2000). Zadnje specifikacije koje je objavio W3C su HTML 4.01, a uskoro se očekuje HTML 5 koji je još u razvoju.

JAVASCRIPT

Najpopularniji jezik za programiranje na webu. Izumio ga je Brendan Eich iz tvrtke Netscape i pojavljuje se u svim preglednicima od 1996. Službeno je standardiziran 1997. od strane ECMA organizacije, a 1998. je odobren kao međunarodni ISO (ISO/IEC 16262) standard.

Javascript omogućuje web stranici interakciju s korisnikom npr. tekst koji se mijenja ovisno o akcijama korisnika, obrada podataka koje je korisnik unio, možemo detektirati koji web preglednik se koristi i proslijediti valjanu stranicu na pregled...

Javascript se izvodi na računalu korisnika (client-side) i zbog sigurnosnih razloga ima neka ograničenja (npr. Nemoguće je spremiti datoteku).

PHP

„Hypertext Preprocesor“. Php je server-side programski jezik (izvodi se na serveru). Podržava mnoge baze podataka kao MySQL, Informix, Oracle, Solid, PostgreSQL, Sybase... Razvio ga je Rasmus Lerdorf kako bi brojao posjete na svojoj stranici. U početku se radilo o skupu pearl skripti, ali je poslije razvijen u C-u i objavljen kao slobodan softver otvorenog koda . Neovisan je o platformi i jako pogodan za razvoj dinamičkih web stranica.

MYSQL

MySQL je besplatan open source sustav za upravljanje bazama podataka. Najčešće je izbor za aplikacije i web stranice otvorenog koda i najčešće se distribuira u serverskim inačicama Linux distribucija iako postoje i verzije za Windowse i MacOS. U svojim početcima MySQL nailazio je na mnogo protivnika jer su mu nedostajale neke osnovne funkcionalnosti definirane SQL standardom. MySQL je optimiziran da bi bio brz nauštrb funkcionalnosti, ali je i vrlo stabilan te ima dobro dokumentirane module i podršku od brojnih programskih jezika. Baze su relacijskog tipa koji se pokazao kao najbolje rješenje za skladištenje velike količine podataka i brzo i kvalitetno dohvaćanje preciznih informacija.

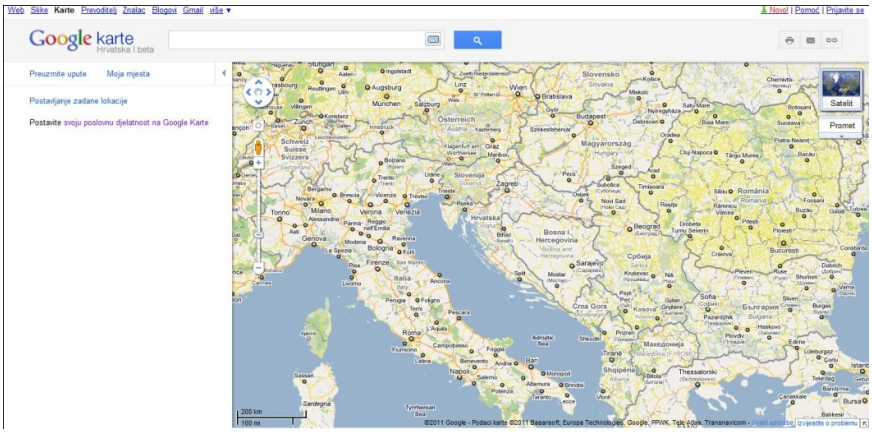
JQUERY

Najpopularnija javascript skripta na webu koju koristi 55% od 10000 najposjećenijih web stranica na webu. Objavio ju je u siječnju 2006. godine John Resig kao softver otvorenog koda. Svrha ove skripte je pojednostavljeno kodiranje na klijentskoj strani, lakši pristup DOM elementima, upravljanje događajima, a omogućava i korištenje lijepih animacija i korisničkog sučelje.

GOOGLE MAPS API

U početku google maps je bila C++ aplikacija koju su razvila dva brata iz Danske Lars i Jens Rasmussen za „Where 2 Tehnologies“. Zamišljena je kao aplikacija koja će se instalirati lokalno na računalo, ali je tvrtka kasnije odlučila da će se raditi o aplikaciji baziranoj na web-u. U listopadu 2004. kompaniju je kupio Google i tako je nastao google maps. Aplikacija je prvi put najavljena u veljači 2005. I imala je podršku samo za Internet Explorer 7.0+ i mozilline web preglednike, ali je ubrzo dodana podrška za sve najpopularnije web preglednike. Tijekom vremena aplikacija se stalno usavršavala kako uslugom tako i sadržajem.

Google maps je web aplikacija koja nudi prikaz karate gotovo cijelog planeta kroz web preglednik. Osim pregleda karte moguće je uključiti i pregled satelitskih slika ili terena. Satelitski pregled ne prikazuje trenutne slike nego slike spremljene na serveru. Uz ove preglede u određenim područjima moguće je uključiti i pogled s ceste (engl. street view) i pregledati kako lokacija izgleda. Također tu su i mnoge druge usluge kao npr. pretraživanje po adresi, planiranje rute (pješice, autom ili javnim prijevozom.)…



Slika 1 Google maps sučelje

U srpnju 2005. Google je lansirao Google Maps API kako bi programerima omogućio integraciju ove usluge u svoje web stranice.

Pomoću ovih alata moguće je postaviti Google Maps na bilo koju web stranicu i preko karata iscrtati svoje podatke. Na taj način korišten je Google Maps u ovom radu. U početku je postojala podrška samo za javascript, ali ubrzo je dodana i podrška za Adobe Flash, usluga za dodavanje statičnih slika karata, generiranje podataka o prometu, za obavljanje geokodiranja te dohvat podataka o nadmorskoj visini. Preko 350 000 web stranica koristi Google maps API. Google maps API je besplatan za sve stranice koje imaju neograničen pristup javnosti.

FANCYBOX

Alat za prikaz galerija slika na web stranicama. Alternativa je klasičnom lightbox-u. Koristi se za prikaz html i multimedijalnog sadržaja u prozoru iscrtanom iznad same web stranice. Za korištenje je potrebna jquery skripta.

CSS

Cascade Style Sheets. Radi se o „stilskom“ jeziku koji se najčešće koristi za opis prezentacijske sheme html dokumenta, ali može se koristit za bilo koju verziju xml dokumenta. Osmišljen je kako bi se odvojio sadržaj od vizualnog dijela te kako bi daljnje promjene bile jednostavnije jer jedna css datoteka može biti povezana s više datoteka sa sadržajem. Također sadržaj se može različito prikazivati ovisni o metodi iscrtavanja (na ekranu ili na papir...).

GPX

GPX (GPS eXchange Format) ja XML shema dizajnirana kako bi služila kao uobičajeni GPS format za GPS aplikacije.

Ovaj format može se koristiti za opis međutočaka, ruta i tragova. Format je otvorenog tipa i nije potrebno plaćati licencu za korištenje. U ovom formatu sprema se lokacija (geografska dužina i širina), nadmorska visina i vrijeme na taj način se može koristiti za izmjenu podataka između GSP naprava i aplikacija. Takav format omogućava korisnicima da pregledavaju njihove tragove na kartama (satelitske snimke, reljefne ili bilo koje druge karte, ili fotografije s geolokacijama).

U GPX-u skup nepovezanih točaka (npr. svi semafori u nekom gradu) se smatra skupinom neovisnih međutočaka. Uređena skupina točaka može biti trag ili ruta. Trag je zapis o stazi koju je netko već prošao dok su rute prijedlozi prolaska u budućnosti. Tako tragovi imaju zapise o vremenu dok rute uglavnom nemaju.

Minimalni potrebni podatci za GPX format su geografska dužina i širina za jednu međutočku. Sve ostalo u zapisu je neobavezno. Od većih proizvođača Humminbird i Garmin koriste GPX format.

Kao mjerne jedinice u GPX formatu koriste se geografska širina i dužina u decimalnom zapisu (s točkom, a ne zarezom), nadmorska visina u metrima. Datum i vrijeme nisu lokalni nego su u UTC formatu uz korištenje ISO 8601 formata.

XAMPP

Besplatni web server otvorenog koda koji je moguće koristiti na različitim platformama (Linux, Windows, MasOS, Solaris). Sastoji se od Apache web servera te MySQL baze podataka, a ima mogućnost interpretirati skripte napisane u php-u i pearl-u. Prvotno namjenjen samo kao pomagalo programerima prilikom razvoja aplikacija nekad se ipak koristi kao web server za prikaz stranica na world wide web-u.

Razvoj aplikacije

Baza Podataka

Baza podataka rađena je pomoću MySQl sustava. Sustav se instalira istovremeno s xampp web serverom i potpuno odgovara zahtjevima aplikacije te je iz toga razloga i izabran.

Baza podataka za ovu aplikaciju sastoji se od šest tablica povezanih relacijama. Svi podatci se spremaju na server kako bismo imali potpunu kontrolu nad njima. Spremaju se podatci o planinarskim objektima u tri tablice:

- OBJECTS (tablica s podatcima o pojedinom objektu)
- OBJECT _IMAGES (tablica koja povezuje pojedini objekt s slikama)
- OBJECT_TYPE (popis tipova objekata)

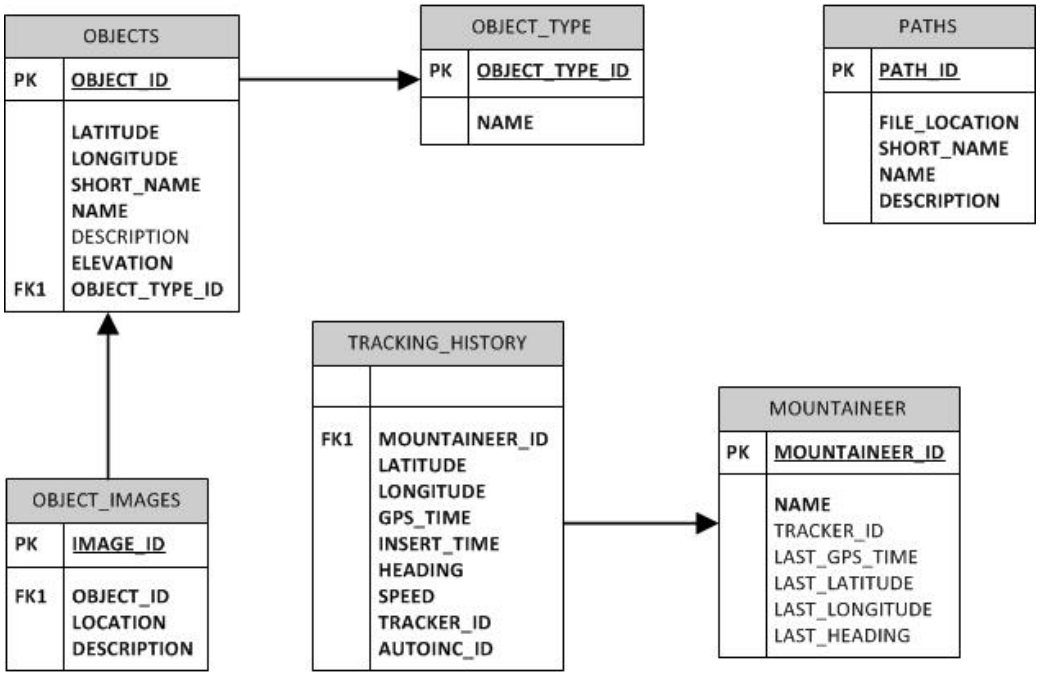
Podatci o planinarima spremaju se u sljedećim tablicama:

- MOUNTAIEER (tablica s podatcima o planinaru)
- TRACKING_HISTORY (tablica s podatcima o kretanju planinara)

Podatci o planinarskim stazama nalaze se u tablici PATHS.

Primjer kreiranja tablice:

```
CREATE TABLE `gisplaninar`.`OBJECTS`
(
      `object_id` int NOT NULL AUTO_INCREMENT PRIMARY KEY ,
      `latitude` float NOT NULL,
      `longitude` float NOT NULL,
      `short_name` varchar(25) NULL,
      `full_name` varchar(100) NOT NULL,
      `description` varchar(250) NULL,
      `elevation` int NOT NULL,
      `object_type_id` int NOT NULL,
    FOREIGN KEY (object_type_id) REFERENCES
OBJECT_TYPE(object_type_id) ON UPDATE CASCADE ON DELETE RESTRICT
) ENGINE = InnoDB CHARACTER SET utf8 COLLATE utf8_general_ci;
```



Slika 2. Model baze podataka

Svi unosi u bazu obavljaju se pomoću php skripti kao i čitanja. Veće količine podataka prenose se aplikaciji u xml formatu koji javascript lako parsira.

Planinarski objekti

Prikaz planinarskih objekata

Kako bismo prikazali geografske informacije o planinarskim lokacijama na karti je prvo potrebno istu prikazati na ekranu. Kartu iscrtavamo unutar div taga koji smo mi postavili preko cijele površine ekrana. Također je potrebno podesiti neke opcije.

```
var myOptions = {
            zoom: 10,
            center: centerPoint,
            mapTypeId: 'terrain'
        }
        map = new
google.maps.Map(document.getElementById("map_canvas"), myOptions);
```

Kako se radi o planinarskoj aplikaciji za vrstu karte izabrali smo terenski prikaz (moguće je još izabrati satelitski prikaz ili cestovnu kartu). Nadalje početnu točku postavili smo u okolici Zagreba (kako bismo pri učitavanju vidjeli Medvednicu i Samoborsko gorje) te uvećanje (engl. zoom) na 10.

Nakon prikaza karte potrebno je na istoj iscrtati markere. Funkcija za postavljanje markera pomoću php skripte iz baze čita informacije o objektima te ih prosljeđuje javascriptu u xml formatu (kod u dodatku A). Za svaki pojedini unos kreira se marker i iscrtava na kartu.

```
var phpscript = "phpscripts/getobjects.php?rand=" +Math.random();
phpscript += "&objecttypes=" + selectedTypes;
phpscript += "&minelevation=" + minelevation + "&maxelevation=" +
maxelevation;
GDownloadUrl(phpscript, function(data) {
    var xml = GXml.parse(data);
    //alert(data);
    var markers =
xml.documentElement.getElementsByTagName("marker");
    for (i = 0; i < markers.length; i++) {
        var naziv = markers[i].getAttribute("naziv");
        var opis = markers[i].getAttribute("opis");
        var vis = parseFloat(markers[i].getAttribute("vis"));
        var tipobjekta = parseInt(markers[i].getAttribute("vrsta"));
        var id = parseInt(markers[i].getAttribute("markerid"));
        var point = new
google.maps.LatLng(parseFloat(markers[i].getAttribute("lat")),
parseFloat(markers[i].getAttribute("lng")));
        createMarker(point, naziv, opis, vis, i, id, tipobjekta);
    }
});
```

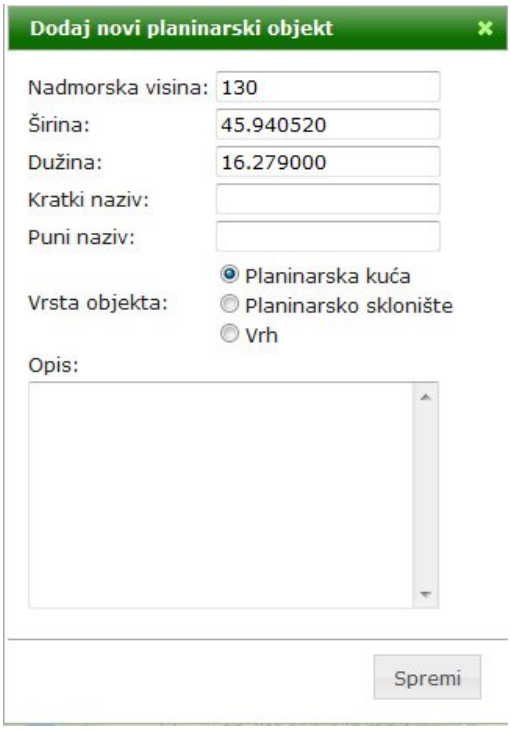
Prilikom stvaranja markera njemu se pridružuje događaj (event.addListener) koji prilikom klika na ikonu objekta otvara informacijski prozor. Unutar njega se nalaze osnovne informacije o objektu te linkovi za prikaz fotografija objekata i dodavanje nove fotografije. Sam marker kreira se pozivom google.maps.Marker.

```
var html = "<b>" + naziv + "</b> ( lat:" + latlng.lat() + ", lon:" +
latlng.lng() + ") <br/>" + "Nadmorska visina: " + vis + "m<br />" +
opis + "<br /><br />";
    html += "<a href=\"javascript:displayFancybox();\">galerija</a>
";
html += "<a href=\"javascript:addObjectImage();\">dodaj sliku</a>";
//određujemo ikonu koja će predstavljati objekt
marker = new google.maps.Marker({map: map, position: latlng, icon:
customIcons[tip]});
markeri[i] = marker;
markerIDs[i] = id;
google.maps.event.addListener(marker, 'click', function () {
    infoWindow.setContent(html);
    infoWindow.open(map, markeri[i]);
    currentmarker = i;
});
```

Dodavanje novog objekta

Ako u alatnoj traci izaberemo dodavanje novog objekta otvara se dijalog za popunjavanje informacija o istom te se na karti pojavljuje pomični marker za izbor lokacije. Nadmorska visina i koordinate se automatski očitavaju s karte pomakom markera i upisuju u odgovarajuća polja.

```
var clickedLocation = event.latLng;
locations.push(clickedLocation);
var positionalRequest = {
    'locations': locations
}
addmarker.setMap(null);
addmarker = new google.maps.Marker({position: clickedLocation, map:
map, icon: icon, shadow: shadow, draggable: true});
elevator.getElevationForLocations(positionalRequest, function
(results, status) {
    if (status == google.maps.ElevationStatus.OK) {
        if (results[0]) {
        document.getElementById("txtaddelevation").value =
results[0].elevation.toFixed(0);
        document.getElementById("txtaddLatitude").value =
results[0].location.lat().toFixed(6);
        document.getElementById("txtaddLongitude").value =
results[0].location.lng().toFixed(6);
    }
```



Slika 3 Dijalog za dodavanje novog objekta

Dodavanje nove vrste objekta nije automatizirano ali je naravno moguće. Prvo je potrebno napraviti ikonu za vrstu objekta. Najlakše je izabrati ikonu s http://mapicons.nicolasmollet.com. Ikonu je potrebno spremiti u „\diplomski\images\Markers“. U datoteci „index.php“

Potrebno je pronaći div ID=“objectLegend“ i dodati novi red u tablicu s ikonom i nazivom:

```
<tr><td>
      <img src="images/Markers/naziv_ikone.png" /><label
style="font-size:larger; vertical-align:middle;"> - Naziv nove vrste
objekta</label></td></tr>
```

U div ID=“select_altitude“ dodati :

```
<input type="checkbox" name="chkselectobject" id="objektnovi naziv"
value="n">Naziv nove vrste objekta</input>
```

n – brojčana vrijednost vrste objekta

U div ID=“ addNewObject“

```
<input type="radio" name="rdbobjectType" value="n" /> nove vrste
objekta <br /></td>
```

Važno je da se vrijednost (value) kod ova dva taga podudaraju jer se radi o ID objekta. Još je u datoteku scripts.js moguće dodati vrijednost odnosno id objekta u polje selectedTypes ako želimo da se objekt prikazuje pri prvom učitavanju te dodati google maps ikonu:

```
customIcons[n] = iconName;
```

Na kraju je potrebno kreirai google maps marker. Marker je moguće kreirati u bilo kojoj javascript datoteci, ali preporučen datoteka je „markerdefinition.js“ .

```
var iconHome = new
google.maps.MarkerImage('images/Markers/naziv_ikone.png',
new google.maps.Size(32, 37),
new google.maps.Point(0, 0),
new google.maps.Point(16, 37));
```

Prvo definiramo lokaciju ikone zatim veličinu, onda poziciju slike unutar grafičkog znaka koji nije potrebno mijenjati te na kraju pomak. Mi pomak postavljano tako da nam mala donja strelica pokazuje na geografsku lokaciju. Lijevi gornji kutak koji pokazuje na lokaciju kao početna vrijednost pomaknemo na polovicu slike „16“ te postavimo da donji rub također pokazuje na lokaciju „37“. Brojeve naravno treba prilagoditi ako ikona nije veličine 32,37 piksela. Ovako na geografsku lokaciju pokazuje sredina donjeg ruba ikone.

Na kraju je u bazu podataka u tablicu object_types potrebno unijeti naziv i id.

Galerija slika

Ako u informacijskom prozoru kliknemo na link za galeriju prikazati će nam se galerija slika objekta ili obavijest da za objekt nema slika. Funkcija prvo pomoću php skripte iz baze čita sve zapise vezane za odabrani objekt. Zatim ih dodaje u html aplikacije i otvara fancybox galeriju za prikaz.

```
GDownloadUrl(phpscript, function(data) {
        var xml = GXml.parse(data);
        var object =
xml.documentElement.getElementsByTagName("object");
        if(object.length == 0)
        {
            alert("Nažalost za odabrani objekt nema slika!");
        }
        else
        {
            var path = object[0].getAttribute("location");
            var description = object[0].getAttribute("description");
            $("#gallery_click").attr("href", path);
            $("#gallery_click").attr("title", description)
            for (i = 1; i < object.length; i++) {
                var object_id = object[i].getAttribute("objectid");
                var path = object[i].getAttribute("location");
                var description =
object[i].getAttribute("description");
                gallery.innerHTML = gallery.innerHTML + "<a
class=\"gallery\" rel=\"gallery_group\" href=\"" + path + "\"
title=\"" + description + "\"></a>";
            }
            $("a.gallery").fancybox({
                titlePosition: 'inside'
            });
            $('#gallery_click').trigger('click');
        }
    });
```

Slike je moguće dodati kroz aplikaciju. Otvaranjem jquery dijaloga prikazuju se input polje za upload slike i unos opisa iste. Klikom na gumb pokreće se php skripta za spremanje podataka na disk i u bazu podataka. Kako bismo osigurali da imena datoteka budu različita, na početak imena datoteke dodajemo slučajan (engl. random) broj. Tako se npr. dvije slike istog objekta s istim nazivom neće spremiti jedna preko druge. Kod se nalazi u Dodatku B. Važno je samo prilikom klika gumba za spremanje povezati div s dijalogom i form tag:

```
 $("#addObjectImage").parent().appendTo($("form#submitform"));
```

Filtriranje prikazanih objekata

Objekte na karti moguće je filtrirati prema vrsti i prema nadmorskoj visini na kojoj se nalaze. Sama logika ugrađena je u funkciju za sliku markera na karti jer se prije kreiranja pojedinog objekta provjerava njegova nadmorska visina i vrsta.

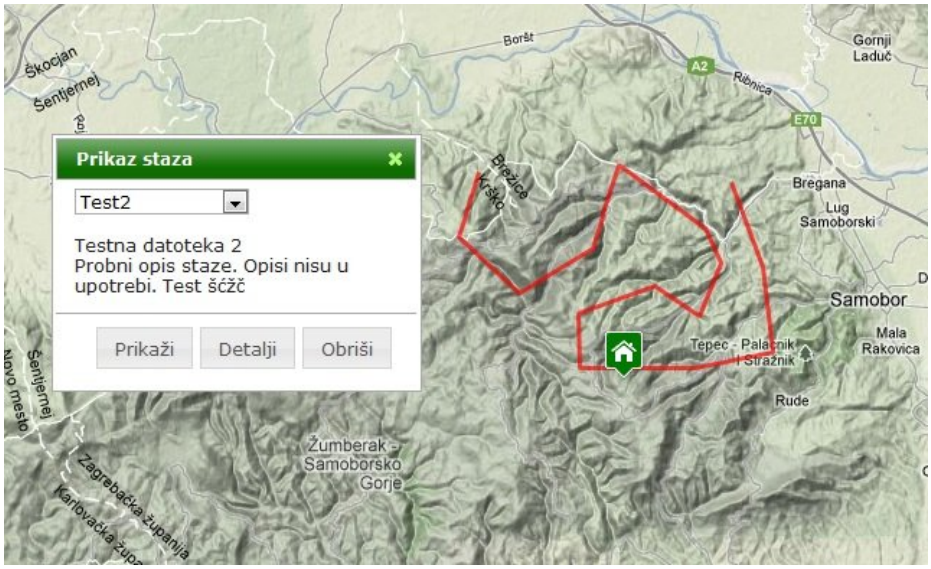
Slika 4 Filtriranje objekata i otvoren prozor za filtriranje prikaza

Planinarske staze

Prikaz planinarskih staza

Planinarske staze prikazujemo pomoću google.maps.Polyline elementa. Parametri ovog elementa su boja, prozirnost i širina linije te naravno polje koordinata. Lokalno smo podatke o pojedinoj stazi spremali na disk u direktorij paths u gpx formatu. Zbog ograničenja javascripta morali smo ekstenziju promijeniti u xml, ali gpx i jeste specijalizirani xml. Svi podatci potrebni za prikaz staze na karti nalaze se u tablici PATHS. Otvaranjem dijaloga za izbor staza možemo iz padajućeg izbornika izabrati željenu stazu te ju iscrtati na karti i dobiti detaljnije informacije o samoj stazi. Otvaranje dijaloga popunjava se padajući izbornik podatcima iz baze na sljedeći način:

```
if($('#ddlselectpath option').size() <= 1)
    {
        GDownloadUrl("phpscripts/getpaths.php", function(data) {
            var xml = GXml.parse(data);
            var paths =
xml.documentElement.getElementsByTagName("paths");
            if(paths.length == 0)
            {
                alert("Nažalost još nije dodana niti jedna staza!");
            }
            else
            {
                for (i = 0; i < paths.length; i++) {
                    var id = paths[i].getAttribute("pathid");
                    var path = paths[i].getAttribute("path");
                    var short_name =
paths[i].getAttribute("shortname");
                    $('#ddlselectpath')
                    .append($("<option></option>")
                    .attr("value",id + "|" + path)
                    .text(short_name));
                }
            }
        });
    }
```



Slika 5 Izbor i prikaz staze

Nakon izbora staze i klika na gumb Prikaži poziva se:

```
function displayPath(path, pathid) {
    if (window.XMLHttpRequest) {
        xmlhttp = new XMLHttpRequest();
    }
    else {// code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET", path, false);
    xmlhttp.send();
    xmlDoc = xmlhttp.responseXML;
    //čitanje markera iz xml datoteke
    var xmlpath = xmlDoc.getElementsByTagName("rtept");
    var points = [];
    xmlhttp.close;
    if(xmlpath.length == 0){
        xmlpath = xmlDoc.getElementsByTagName("trkpt");
    }
    for (i = 0; i < xmlpath.length; i++) {
        points[i] = new
google.maps.LatLng(parseFloat(xmlpath[i].getAttribute("lat")),
parseFloat(xmlpath[i].getAttribute("lon")));
    }
    var pathline = new google.maps.Polyline({
        path: points,
        strokeColor: "#FF0000",
        strokeOpacity: 0.7,
        strokeWeight: 3
    });
    pathline.setMap(map);
    polylines[polylines.length] = pathline;
    polylinesIDs[polylinesIDs.length] = pathid;
}
```

Za svaki element iz xml-a koji ima tag rtept (route point) ili trkpt (track point) dodaje se jedna točka u polje točaka. Onda kreiramo google maps polyline koji iscrtavamo na karti te poligon spremamo u globalnu varijablu kako bismo ga mogli obrisati. Ako kliknemo na gumb detalji iz baze se čitaju detalji za odabrani poligon i prikazuju u dijalogu. Predviđeno je da se u detalje upiše opis najvažnijih točaka staze.

Track point je točka generirana kreiranjem traga i korištenjem gps uređaja, dok je route point točka nastala planiranjem rute i njome nije netko prošao nego se koristi za planiranje.

Dodavanje planinarske staze iscrtavanjem s karte

Planinarskih ruta hrvatskih planina u bilo kojem gps formatu nema mnogo pa je ponuđeno i ucrtavanje staze unutar aplikacije. Otvaranje dijaloga moguće je upisati kratki naziv staze (preporučeno početna i/ili završna lokacija), pun naziv koji sadrži glavne točke rute i detaljan opis s trajanjem hodanja i opisom mjesta na kojima treba paziti da se ne bi skrenulo sa staze.

Prvo je potrebno inicijalizirati postavke i google maps polyline:

```
function initialiseaddroute() {
    var polyOptions = {
        strokeColor: '#00AA00',
        strokeOpacity: 1.0,
        strokeWeight: 3
    }
    poly = new google.maps.Polyline(polyOptions);
    poly.setMap(map);

    // dodaj event za klik na kartu
    google.maps.event.addListener(map, 'click', addpolylinepoint);
}
```

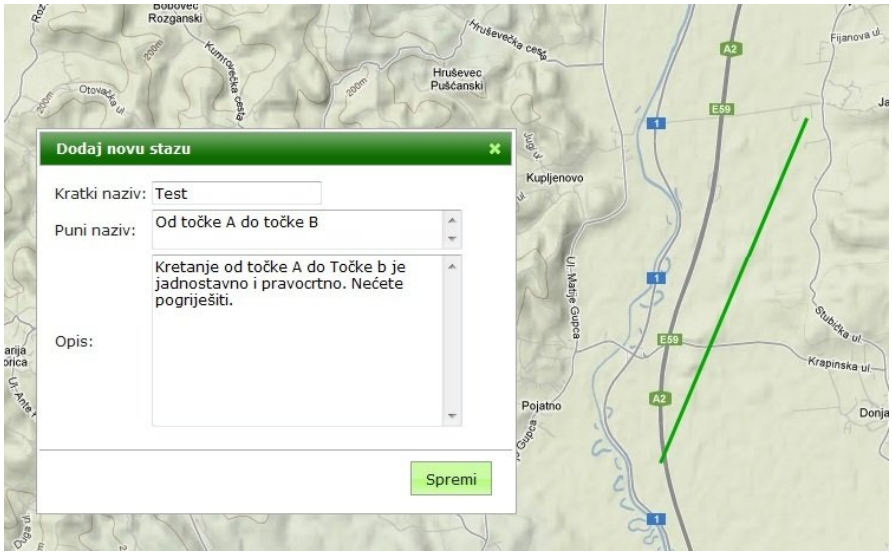
Svakim klikom na kartu u polyline se dodaje jedna točka:

```
function addpolylinepoint(event) {
    var path = poly.getPath();
    path.push(event.latLng);
}
```

Dodavanje točke ona se automatski iscrtava na karti. Kada smo Završili unošenje točaka i popunili potrebna polja u dijalogu klikom na gumb spremi pokrećemo:

```
function addRoute() {
    poly.setMap(null);
    var path = poly.getPath();
    var pointslat = [];
    var pointslon = [];
    var polypoints = path.b;
    for (i = 0; i < polypoints.length; i++) {
        pointslat[i] = polypoints[i].lat();
        pointslon[i] = polypoints[i].lng();
    }
    var phpscript = "phpscripts/addroutemap.php";
    var shortname =
document.getElementById("addroutemapshort").value;
    var name = document.getElementById("addroutemapname").value;
    var routedescription =
document.getElementById("txtpathmapdescription").value;
    phpscript += "?shortname=" + shortname + "&name=" + name +
"&description=" + routedescription;
    phpscript += "&latitudes=" + pointslat.toString() +
"&longitudes=" + pointslon.toString();
    google.maps.event.clearListeners(map, 'click');
    phpscript = encodecroatianurlchars(phpscript);
    GDownloadUrl(phpscript, function(data) {
       alert(data);
    });
    document.getElementById("ddlselectpath").options.length = 1;
}
```

Pročitamo sve točke iz polja te opis i naziv staze i proslijedimo ih php datoteci koja nakon toga od podataka generira potrebnz gpx datoteku te je spremi na disk i doda potrebne podatke u bazu u tablicu PATHS. Php skripta je u dodatku C.



Slika 6 Dodavanje planinarske staze iscrtavanjem na karti

Dodavanje planinarske staze pomoću gpx datoteke

Dodavanje planinarske staze učitavanjem gpx datoteke vrlo je slično ucrtavanju s karte. Razlika je jedino što se gpx datoteka ne generira nego se samo vrši učitavanje (engl. upload) pomoću input polja vrste „file“. Unose se iste informacije o staze: kratki naziv, naziv i opis. Također kako se radi o spremanju datoteke klikom na gumb potrebno je izvršiti

```
$("#addroutegpx").parent().appendTo($("form#addroutegpxform"));
```

kao i kod dodavanja slike u galeriju objekta.

Prikaz planinara

Trenutne lokacije

Kao i za prikaz planinarskih objekata i za prikaz trenutnih lokacija planinara koristili smo google maps markere. Za ikonu je odabrana sličica koja podsjeća na planinarsku markaciju (žuti krug sa crvenim rubom).

```
var iconHiker = new
google.maps.MarkerImage('images/Markers/hiker1.png',
new google.maps.Size(16, 16),
new google.maps.Point(0, 0),
new google.maps.Point(8, 8));
```

Klikom na prikaz planinara iz baze se čitaju informacije o planinarima (naziv i geografski podatci) te se slično kao i lokacije planinarskih domova podatci pomoću php skripte prosljeđuju u javaskript kod u xml formatu. Zatim je iscrtavaju markeri na potrebne lokacije.

Slika 7 Prikaz lokacija planinara

```
function createhiker(latlng, name, i, id) {
    var html = "<b>" + name + "</b> <br />( lat:" + latlng.lat() +
", lon:" + latlng.lng() + ") <br />";
    //određujemo ikonu koja će predstavljati objekt
    hiker = new google.maps.Marker({map: map, position: latlng,
icon: iconHiker});
    hikeri[i] = hiker;
    hikerIDs[i] = id;
    google.maps.event.addListener(hiker, 'click', function () {
        infoWindow.setContent(html);
        infoWindow.open(map, hikeri[i]);
        currenthiker = i;
    });
}
```

Prijeđeni put

Prijeđeni put planinara prikazuje se pomoću google maps ployline elementa. Kako aplikacija prikazuje putanje više planinara trag svakog prikazuje se drugom bojom. Početna boja bira se u sustavu nijanse, svjetline i zasićenja. Željeli smo dobiti pune boje pa smo zasićenje postavili na 1, svjetlinu na 0,5 kako boje ne bi bile ni pretamne niti blijede. Nijansu za prvu boju birali smo pomoću slučajnog broja i onda ju za svakog sljedećeg planinara povećavali za 1/12 spektra. Sada boju valja prebaciti u RGB oblik.

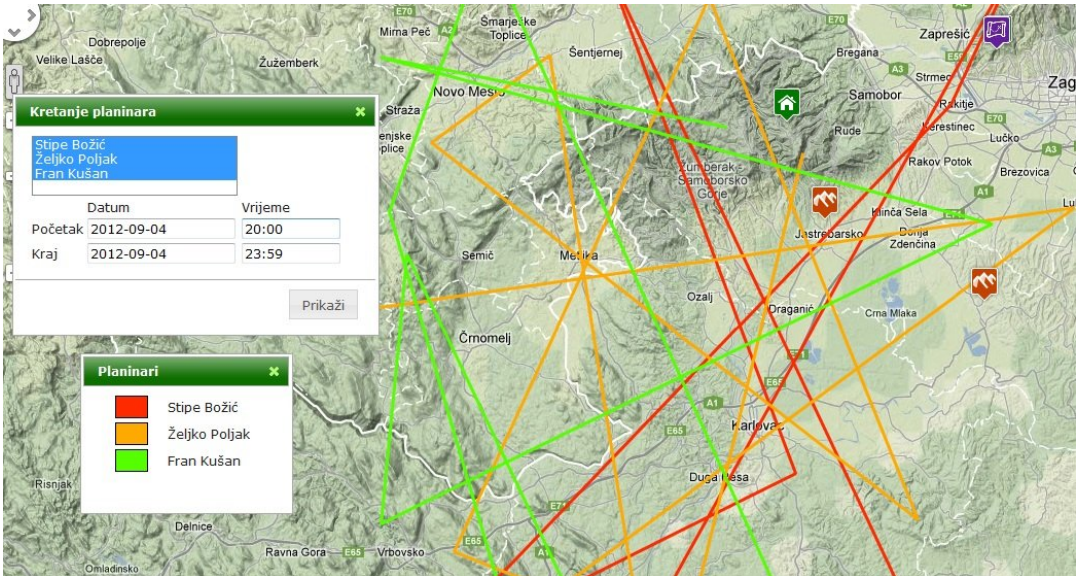
```
function getcolor(hue)
{
    var h = hue;
    var s = 1;
    var l = 0.5;
    if (s === 0) { return [l, l, l]; }
    var temp1, temp2;
    if (l < 0.5) { temp2 = l * (1.0 + s); }
    else { temp2 = l + s - l * s; }
    temp1 = 2.0 * l - temp2;
    var r, g, b;
    var getCol = function (t1, t2, t3) {
        var col;
        if (t3 < 0) { t3 = t3 + 1.0; }
        else if (t3 > 1) { t3 = t3 - 1.0; }
        if (6 * t3 < 1) { col = t1 + (t2 - t1) * 6.0 * t3; }
        else if (2 * t3 < 1) { col = t2; }
        else if (3 * t3 < 2) { col = t1 + (t2 - t1) * (2.0 / 3.0 -
t3) * 6.0; }
        else { col = t1; }
        return col;
    };
    r = getCol(temp1, temp2, h + 1.0 / 3.0);
    g = getCol(temp1, temp2, h);
    b = getCol(temp1, temp2, h - 1.0 / 3.0);
```

```
    return "rgb(" + Math.round(r * 255) + "," + Math.round(g * 255)
+ "," + Math.round(b * 255) + ")";
}
```

Ova funkcija je općenita i iako mi koristimo uvijek isto zasićenje i svjetlinu zbog buduće upotrebe ostavili smo sve slučajeve.

Iz baze dohvaćamo podatke za odabrane planinare te biramo vremensko razdoblje za koje želimo prikazati putanje (php skripta u dodatku D) i iscrtavamo linije na karti:

```
GDownloadUrl(phpscript, function(data) {
    var xml = GXml.parse(data);
    var hikers = xml.documentElement.getElementsByTagName("hiker");
    var points = [];
    var pathline;
    clearhikertable();
    for(var j=0; j < selectedhikers.length; j++)
    {
        points = [];
        var index = 0;
        for (i = 0; i < hikers.length; i++) {
            var hikerid =  hikers[i].getAttribute("hikerid");
            if(hikerid == selectedhikers[j])
            {
                points[index] = new
google.maps.LatLng(parseFloat(hikers[i].getAttribute("lat")),
parseFloat(hikers[i].getAttribute("long")));
                index++;
            }
        }
        //get color
        startcolor += j*1.0/12;
        if(startcolor > 1) startcolor-= 1;
        var color = hue2rgb(startcolor);
        pathline = new google.maps.Polyline({
            path: points,
            strokeColor: color,
            strokeOpacity: 1,
            strokeWeight: 3
        });
        pathline.setMap(map);
        $("#hikerstable").last().append("<tr><td><div
class='hikerColor' style='background-color:" + color + ";
width:30px; height:20px;'></div></td><td>" + selectednames[j] +
"</td></tr>");
        hikerpolylines[j] = pathline;
    }
});
```



Slika 8 Prikaz putanja planinara

Mjerenja

Mjerenje udaljenosti

Otvaranje iskočnog (engl. popup) prozora također se aktivira događaj klika na kartu. Prilikom klika iscrtava se linija slično kao i kod iscrtavanja staza samo se točke dodaju pri svakom kliku. Prilikom očitanja geografske širine i duljine računa se udaljenost između izabrane i prijašnje točke. Prilikom proračuna važno je paziti na činjenicu da se meridijani približuju jedan drugome kad se pomičemo sjevernije. Zato treba svaki puta računati duljinu jednog stupnja geografske dužine po formuli:

$$\Delta^1_{LONG} = \frac{\pi a cos\emptyset}{180(1 - e^2 sin^2\emptyset)^{1/2}}$$

Geografska širina ne mijenja se tako drastično pa se uzima prosječna vrijednost za 45° sjeverne geografske širine.

```
function calculateLonLength(lon) {
    //duljina jednog stupnja u metrima
    var R = 6378.137;
    var rad = 0.0174532925199433;
    return 1 / 180 * R * Math.PI * Math.cos(lon * rad);
}
function lineLength(lastPoint) {
    if (oldPoint == undefined) {
        oldPoint = lastPoint;
        return 0;
    }
    else {
        var sLat, sLon, fLat, fLon;
        sLat = oldPoint.lat();
        sLon = oldPoint.lng();
        fLat = lastPoint.lat();
        fLon = lastPoint.lng();
        var x = (fLat - sLat) * 111000;
        var y = (fLon - sLon) * calculateLonLength((fLat + sLat) /
2) * 1000;
        d = Math.sqrt(x * x + y * y);
    }
    oldPoint = lastPoint;
    return d;
}
```

Mjerenje površine

Kao i kod mjerena udaljenosti otvaranjem dijaloga aktivira se događaj klika na karti te se inicijalizira poligon. Poligon je vrlo sličan polyline-u, ali je razlika u tome što se početna i završna točka spajaju i boja se unutrašnjost .

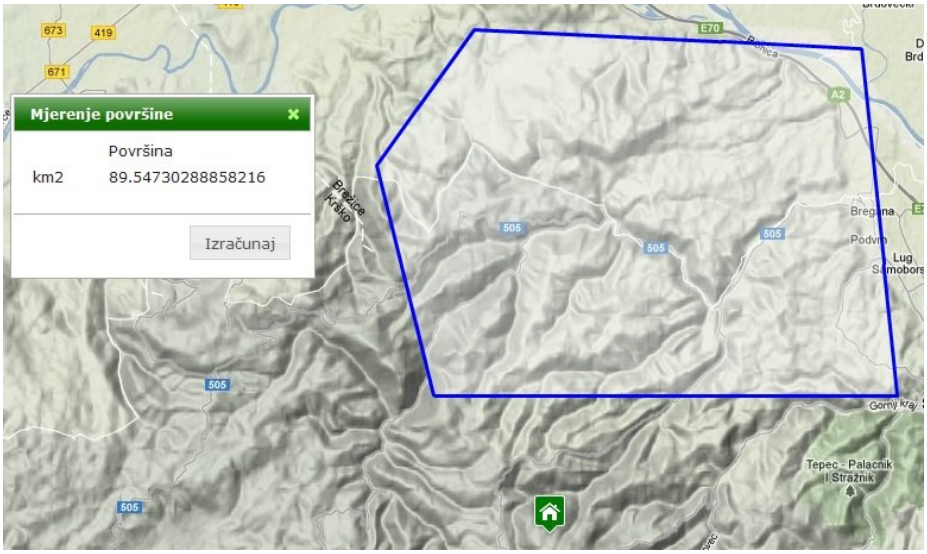
```
function initialisesurface() {
    var polygonOptions = {
        strokeColor: "#0000FF",
        strokeOpacity: 1,
        strokeWeight: 3,
        fillColor: "#FFFFFF",
        fillOpacity: 0.35
    }
    polygon = new google.maps.Polygon(polygonOptions);
    polygon.setMap(map);
    // dodaj event za klik na kartu
    google.maps.event.addListener(map, 'click', addpolygonpoint);
}
```

Svakim klikom dodaje se nova točka u poligon, ali se površina računa tek klikom na gumb Izračunaj. Površine se računaju po sljedećoj formuli:

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)$$

Ispravan rezultat dobiva se ako nema presjecanja linija poligona.

```
function surfacesize(ploygonpath) {
    var a = 0;
    var LatLength = 111.132;
    var LonLength = 0;
    var sLat, sLon, fLat, fLon;
    for (var i = 0; i < ploygonpath.length; i++) {
        LonLength = LonLength + ploygonpath[i].lat();
    }
    LonLength = calculateLonLength(LonLength / ploygonpath.length)
    for (i = 0; i < ploygonpath.length; i++) {
        sLat = ploygonpath[i].lat();
        sLon = ploygonpath[i].lng();
        if (i == ploygonpath.length - 1) {
            fLat = ploygonpath[0].lat();
            fLon = ploygonpath[0].lng();
        }
        else {
            fLat = ploygonpath[i + 1].lat();
            fLon = ploygonpath[i + 1].lng();
        }
        a = a + ((sLat * fLon) - (fLat * sLon)) * LatLength *
LonLength;
    }
    a = Math.abs(a / 2);
    return a;
}
```



Slika 9 Mjerenje površine

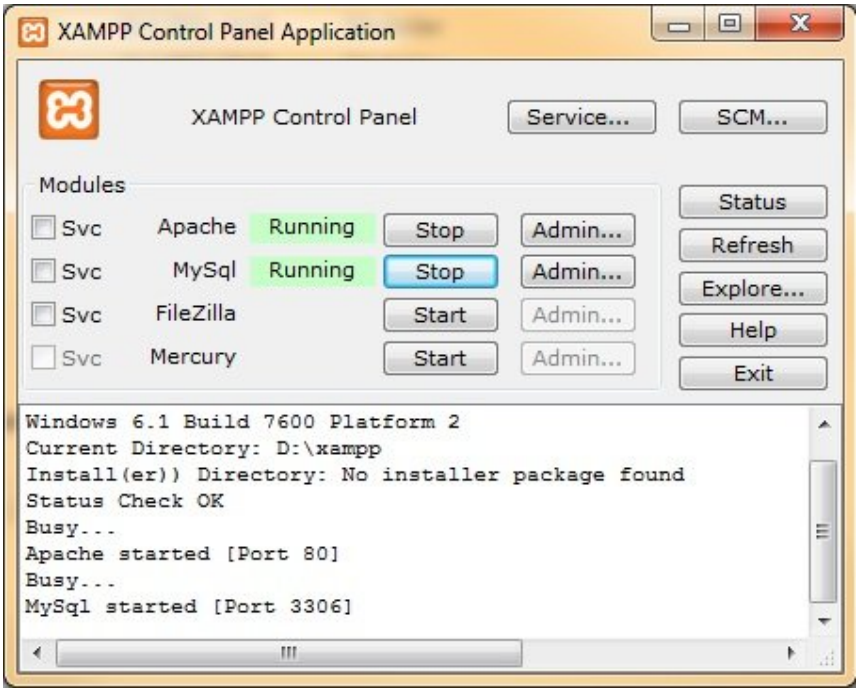
Upute za instalaciju

Instalacija web servera

Instalacija aplikacija nije potrebna jer se radi o web aplikaciji, ali ako se aplikaciju izvodi lokalno na računalu potrebno je instalirati web server. Prilikom razvoja korišten je xampp koji je priložen s CD-om uz rad. Priložena je verzija za windows operativne sustave ali moguće je preuzeti i verzije za ostale operacijske sustave na web stranici:

http://www.apachefriends.org/en/xampp.html

Instalacija za windows sustave je najjednostavnija ako se koristi installer, ali moguće je preuzeti i zip datoteku. Nije preporučeno potrebne datoteke raspakirati/instalirati u Program Files direktorij jer on ima određena ograničenja koja mogu ometati rad web servera, nego je najbolje server postaviti u npr. „c:/xampp“. Prije pokretanja web servera potrebno ga je konfigurirati pokretanjem datoteke „setup-xampp.bat“ u mapi „xampp“. Server pokrećemo dvostrukim klikom na „xampp-control.exe“ i otvara se sljedeći prozor:



Slika 10 Kontrolni prozor xampp web servera

Potrebno je pokrenuti Apache i MySQL usluge.

Ako koristite još neki web server ili neki drugi program koji koristi port 80 potrebno je dodatno podesiti portove. U datoteci „httpd.conf“ koja se nalazi „..\xampp\apache\conf“ potrebno je promijeniti liniju „Listen 80“ u „Listen 81“ ili neki drugi port i „ServerName localhost:80“ u „ServerName localhost:81“. Sada se naš web server nalazi na portu 81. Također u datoteci „httpd-ssl.conf“ na lokaciji „..\xampp\apache\conf\extra“ promijeniti „Listen 443“ u „Listen 442“ ili neki drugi port.

Sada je cijeli sadržaj direktorija „diplomski potrebno iskopirati u „..\xampp\htdocs“. Aplikaciji se pristupa upisivanjem „localhost/diplomski“ u prostor za adresu web preglednika. Ako je promjenjen port onda se upisuje „localhost:81/diplomski“.

Kreiranje baze podataka

Sada je potrebno kreirati bazu podataka. Koristiti ćemo phpmyAdmin servis koji dolazi s xampp web server paketom. U Web preglednik upisujemo „localhost“ (dodajemo port ako smo ga promijenili) i s lijeve strane izabiremo phpmyAdmin. Odmah kreiramo novu bazu podataka naziva gisplaninar, a za uspoređivanje odaberemo utf8_general_ci. Kada smo kreirali bazu i klikom na njeno ime s lijeve strane došli do postavki iste možemo kreirati potrebne tablice. Tablice je najlakše kreirati koristeći priloženu datoteku „sqlDiplomski_ CreateTables.sql“ koja se nalazi na priloženom CD-u. Baza je ovime kreirana, ali u njoj nema podataka koje bi aplikacija prikazivala. Testne podatke moguće je dodati klikom na tab Uvoz i izborom datoteke „sample_data.sql“ koja se nalazi na priloženom CD-u.

Zaključak

Tijekom ostvarivanja ovog rada kreirana je web aplikacija za prikaz i unos geografskih podataka planinara koristeći google maps API. Težili smo stvoriti programsku aplikaciju temeljenu na otvorenom kodu i besplatnim programskim alatima. Početna ideja bila je sadržaj nekoliko knjiga koje opisuje Hrvatske planina pretočiti u dinamičnu i korisnu aplikaciju.

Aplikacija ima mogućnost prikaza planinarskih objekata koje smo podijelili u osnovne kategorije. Korisnik može pregledati lokacije pojedinog objekta i dobiti osnovne informacije o njemu. Također ponuđen je prikaz slika, a korisnik ima mogućnost i dodavanja novog sadržaja.

Također moguće je prikazivati planinarske staze na karti te ispisati osnovne informacije o pojedinoj stazi.

Nadalje unutar aplikacije moguće je prikazivati lokacije i prijeđene putove planinara koji na server šalju svoje gps podatke. Modul za unos podataka u bazu nije napravljen jer nije dio ovoga rada, ali je napravljena priprema i prikazuju se slučajnio odabrani podatci kao lokacije planinara.

Napravljeni su i alati za mjerenje površine i udaljenosti.

Aplikacija je intuitivna i nije potrebno nikakvo dodatno školovanje da bi se koristila.

Moguća su i daljnja proširenja i razvoj aplikacije. Uobičajno se planinarski domovi i staze grupiraju po gorjima u kojima se nalaze pa bi i naša aplikacija u budućoj nadogradnji mogla prikazivati zone tako da korisnik može birati prikaz planinarskih domova i staza samo u odabranoj zoni. Također u radu je fokus bio na funkcionalnosti, ali objavom aplikacije javlja se i pitanje sigurnosti i obrane od napada. Nadalje svjedoci smo velikog širenja mobilnih platformi i njihovih sve većih mogućnosti te ne treba zanemariti korisnike istih i prilagoditi aplikaciju kako bi ju i oni mogli ugodno koristiti. U sustavu za praćenje planinara moguće se raznolike nadogradnje i prilagodbe koje bi ovisile o specifičnom slučaju. Npr. u slučaju planinarske

utrke mogla bi se dodati mogućnost prikaza prijeđene staze u stvarnom vremenu.

Brojne su i vrlo slične primjene gdje su opisane tehnologije primjenjive npr. u praćenju vozila (brodovi, automobili, avioni), praćenju izgubljenog ili ukradenog mobitela i uz prilagodbe i daljnji razvoj ova aplikacija mogla bi poslužiti kao osnova za razvoj takvih programskih aplikacija.

Literatura

1. Željko Poljak: Hrvatske planine
2. Alan čaplar: Planinarski vodič po Hrvatskoj
3. Luka Abrus: Izrada Weba - abeceda za webmastere
4. http://www.w3schools.com/
5. http://jquery.com/
6. http://jqueryui.com/
7. http://fancybox.net/
8. http://mapicons.nicolasmollet.com
9. http://www.htmldog.com/articles/suckerfish/dropdowns/
10. http://www.apachefriends.org/en/xampp.html
11. http://econym.org.uk/gmap/basic7.htm
12. https://developers.google.com/maps/documentation/javascript/basics.html
13. http://local.wasp.uwa.edu.au/~pbourke/texture_colour/convert/
14. http://dev.mysql.com/

SAŽETAK

(WEB APLIKACIJA ZA PRIKAZ I UNOS GEOGRAFSKIH PODATAKA PLANINARA**)**

U ovom radu napravljena je web aplikacija za prikaz i unos geografskih podataka planinara. Prikazuju se lokacije planinarskih domova i informacije o njima, Planinarske staze i lokacije planinara te staze kojima su prošli. Također moguće je mjeriti udaljenosti i površine. Korištene su tehnologije otvorenog koda kako bi razvoj bio što jednostavniji i dostupniji.

Osnova je Google maps API, javascript (jquery), php i MySQL, a aplikacija se izvodi na apache web serveru.

ABSTRACT

In this study we have made a web application to display and input geographic data of mountaineers. Locations of mountain homes and information about them, hiking trails and hiking locations and paths that are passed are shown. It is also possible to measure distances and areas. Were used open-source technologies because of easier and more accessible future development.

The basis of development was Google Maps API, Javascript (jQuery), PHP and MySQL, and application is running on the Apache Web server.

Dodatak

Dodatak A

Čitanje podataka o objektima.

```
<?php

require("dbinfo.php");

// stvaranje xml-a

$dom = new DOMDocument("1.0", 'utf-8');
$node = $dom->createElement("markers");
$parnode = $dom->appendChild($node);

$selectedtypes = $_GET["objecttypes"];
$minelevation = $_GET["minelevation"];
$maxelevation = $_GET["maxelevation"];

$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {  die('Not connected : ' . mysql_error());}

//postavke za hrbatske znakove
mysql_query('SET NAMES UTF8');
// odabir baze podataka
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
  die ('Can\'t use db : ' . mysql_error());
}

$query = "SELECT object_id, latitude, longitude, short_name,
full_name, description, elevation, object_type_id
FROM OBJECTS WHERE elevation > $minelevation AND elevation <
$maxelevation AND OBJECT_TYPE_ID in($selectedtypes)";
$result = mysql_query($query);
if (!$result) {
  die('Invalid query: ' . mysql_error());
}
header("Content-type: text/xml");
while ($row = @mysql_fetch_assoc($result)){
  $node = $dom->createElement("marker");
  $newnode = $parnode->appendChild($node);
  $newnode->setAttribute('markerid',$row['object_id']);
  $newnode->setAttribute('lat',$row['latitude']);
  $newnode->setAttribute('lng',$row['longitude']);
  $newnode->setAttribute('opis',$row['description']);
  $newnode->setAttribute('kratkinaziv',$row['short_name']);
  $newnode->setAttribute('naziv',$row['full_name']);
  $newnode->setAttribute('vis',$row['elevation']);
  $newnode->setAttribute('vrsta',$row['object_type_id']);
}

echo $dom->saveXML();

?>
```

Dodatak B

Upload slike i spremanje podataka u bazu

```
<?php
require("dbinfo.php");
//citanje podataka iz formulara i spremanje u varijable
$description = $_POST["imagedesc"];
$id = $_POST["addimageobject_id"];
$filetype = $_FILES["addimagefile"]["type"];
$filename = $_FILES["addimagefile"]["name"];
$target_path = "gallery/$id/$filename";
if ($_FILES["addimagefile"]["error"] > 0)
   {
   echo "Apologies, an error has occurred.";
   echo "Error Code: " . $_FILES["addimagefile"]["error"];
   }
else
{
    if($filetype=="image/x-png" || $filetype=="image/pjpeg" ||
$filetype=="image/gif" || $filetype=="image/jpeg")
    {
        if(!is_dir("../gallery/$id"))
        {
            mkdir("../gallery/$id");
        }
        if(move_uploaded_file($_FILES['addimagefile']['tmp_name'],
"../$target_path")) {
            $con = mysql_connect('localhost', $username, $password);
            if (!$con)
            {
            die('Could not connect: ' . mysql_error());
            }
            mysql_query('SET NAMES UTF8');
            mysql_select_db($database, $con);

            //unošenje podataka iz formulara u bazu
            $sql = " INSERT INTO `gisplaninar`.`OBJECT_IMAGES` (
            `object_id`, `location`, `description`
            )
            VALUES ('$id', '$target_path', '$description')";
            if (!mysql_query($sql,$con))
            {
                echo("Greška prilikom spremanja podataka!");
                die('Error: ' . mysql_error());
            }
            else
            {
                echo "<script type='text/javascript'>\n";
                echo "window.location = '../index.php'; ";
                echo "alert('Podatci su uspješno spremljeni');";
                echo "</script>";
            }
        }
        else{
            echo "There was an error uploading the file, please try
again!";
        }
    }
```

```
    else
    {
        echo "<script type='text/javascript'>\n";
        echo "alert('Dozvoljen je upload samo slika sljedećih
formata: gif, jpeg, jpg, png');";
        echo "window.location = '../index.php'; ";
        echo "</script>";
    }
}

?>
```

Dodatak C

Spremanje planinarske staze

```
<?php

require("dbinfo.php");

//citanje podataka iz formulara i spremanje u varijable
$description = $_GET["description"];
$shortname = $_GET["shortname"];
$name = $_GET["name"];
$latitudes = $_GET["latitudes"];
$longitudes = $_GET["longitudes"];

$dom = new DOMDocument("1.0", 'utf-8');
$node = $dom->createElement("gpx");
$node->setAttribute("xmlns", "http://www.topografix.com/GPX/1/1");
$node->setAttribute("creator", "GisPlaninar");
$gpxnode = $dom->appendChild($node);

$node = $dom->createElement("metadata");
$metanode = $gpxnode->appendChild($node);

$node = $dom->createElement("link");
$linknode = $metanode->appendChild($node);
$linknode->setAttribute('href','Web lokacija jos nije definirana');
$node = $dom->createElement("text");
$linknode->appendChild($node);
$text = $dom->createTextNode("GIS Planinar");
$node->appendChild($text);

$lat = explode(",", $latitudes);
$long = explode(",", $longitudes);

$node = $dom->createElement("rte");
$rtenod= $gpxnode->appendChild($node);

$today = date("Y\-m\-d\TG\:m\:s");

for($i=0;$i<sizeof($lat);$i++)
{
    $node = $dom->createElement("rtept");
    $rteptnode = $rtenod->appendChild($node);
    $node->setAttribute("lat", $lat[$i]);
    $node->setAttribute("lon", $long[$i]);
```

```
    $node = $dom->createElement("time");
    $timenode = $rteptnode->appendChild($node);
    $time = $dom->createTextNode($today);
    $timenode->appendChild($time);
}

$ran = rand () ;
$target_path = "paths/$ran$shortname.xml";

$content = $dom->saveXML();
//spremanje u datoteku
$fhandle = fopen("../$target_path","w");
fwrite($fhandle,$content);
fclose($fhandle);


$con = mysql_connect('localhost', $username, $password);
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_query('SET NAMES UTF8');
mysql_select_db($database, $con);
$sql = " INSERT INTO `gisplaninar`.`PATHS` (
`FILE_LOCATION`, `SHORT_NAME`, `NAME`, `DESCRIPTION`
)
VALUES ('$target_path', '$shortname', '$name', '$description')";
if (!mysql_query($sql,$con))
{
    echo("Greška prilikom spremanja podataka!");
    die('Error: ' . mysql_error());
}
else
{
    echo 'Podatci su uspješno spremljeni';
}

?>
```

Dodatak D

Čitanje podataka o prijeđenom putu za zadane planinare

```
<?php

require("dbinfo.php");

$dom = new DOMDocument("1.0", 'utf-8');
$node = $dom->createElement("hikers");
$parnode = $dom->appendChild($node);

$ids = $_GET["ids"];
$starttime = $_GET["starttime"];
$endtime = $_GET["endtime"];

$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {  die('Not connected : ' . mysql_error());}

mysql_query('SET NAMES UTF8');
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
  die ('Can\'t use db : ' . mysql_error());
}
$query = "SELECT `MOUNTAINEER_ID`, `INSERT_TIME`, `GPS_TIME`,
`LATITUDE`, `LONGITUDE`, `HEADING`, `SPEED` FROM `tracking_history`
WHERE `MOUNTAINEER_ID` IN($ids) AND `GPS_TIME` BETWEEN '$starttime'
AND '$endtime'";
$result = mysql_query($query);
if (!$result) {
  die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// prolazimo kroz cijelu tablicu i za svaki red upisujemo podatke u
xml

while ($row = @mysql_fetch_assoc($result)){

  $node = $dom->createElement("hiker");
  $newnode = $parnode->appendChild($node);
  $newnode->setAttribute('hikerid',$row['MOUNTAINEER_ID']);
  $newnode->setAttribute('gpstime',$row['GPS_TIME']);
  $newnode->setAttribute('lat',$row['LATITUDE']);
  $newnode->setAttribute('long',$row['LONGITUDE']);
  $newnode->setAttribute('heading',$row['HEADING']);
  $newnode->setAttribute('speed',$row['SPEED']);
}
echo $dom->saveXML();

?>
```