

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA
MOBILNA APLIKACIJA ZA GLASOVNO VODENJE PO VELENJU

Tematsko področje: RAČUNALNIŠTVO

Avtorji:

Tomaž Čede, 3. letnik

Jon Rojnik Goršek, 3. letnik

Tilen Hostnik, 3. letnik

Mentorja:

Gregor Hrastnik, univ. dipl. inž.

Samo Železnik, inž.

Velenje, 2020

Raziskovalna naloga je bila opravljena na Elektro in računalniški šoli Šolskega centra Velenje, 2020.

Mentorja: Gregor Hrastnik, univ. dipl. inž., Samo Železnik, inž.

Datum predavitve: marec 2020

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD ŠC Velenje, šolsko leto 2019/2020

KG GPS / Velenje / orientacija / vodenje / način vodenja

AV Tomaž Čede / Tilen Hostnik / Jon Rojnik Goršek

SA HRASTNIK, Gregor / ŽELEZNIK, Samo

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2020

LI 2020

IN MOBILNA APLIKACIJA ZA GLASOVNO VODENJE PO VELENJU

TD Raziskovalna naloga

OP

IJ SL

JI sl / en

AI Raziskovalno nalogo na temo glasovnega vodenja nam je predlagal naš mentor, idejo smo sprejeli in si zamislili, kako bi lahko Velenje naredili bolj dostopno turistom. V aplikaciji bi si uporabnik izbral določeno pot, ki je že vnaprej programirana. Nato bi telefon pospravili v žep in bi ga aplikacija skozi slušalke vodila preko glasovnih posnetkov. Na poti bi mu tudi pripovedovala o kakšnih bližnjih znamenitostih.

KEY WORDS DOCUMENTATION

ND ŠCV, šolsko leto 2019/2020

CX GPS / Velenje / orientation / guidance / a way of leading

AU Tomaž Čede / Tilen Hostnik / Jon Rojnik Goršek

AA HRASTNIK, Gregor / ŽELEZNIK, Samo

PP 3320 Velenje, SLO, Trg mladosti 3

PB ŠC Velenje, Elektro in računalniška šola, 2020

PY 2020

TI MOBILE APP WITH VOICE GUIDANCE THROUGH VELENJE

DT RESEARCH WORK

NO

LA SL

AL sl / en

AB Research on the topic of navigation with the use of voice was already suggested by our mentor, and we came up with the idea only after a long reflection on how we could make Velenje more accessible to tourists. In the application, the user would choose a specific path already programmed in advance. Then the phone would be put away into our pockets and the app would guide us through headphones with the use of the recorded voices. On the way, It would also tell us about any nearby attractions.

KAZALO KRATIC

IDE – Integrated Development Environment

GPS – Global Positioning System

AR – Augmented Reality

API – Application Programmable Interface

KAZALO VSEBINE

1. UVOD	1
1.1 Hipoteze	1
2. PREGLED STANJA TEHNIKE	2
2.1 Mobilne aplikacije	2
2.2 GPS	3
2.3 Podobne aplikacije	4
Google Zemljevidi	4
Apple Maps	5
3.METODOLOGIJA	7
3.1 Izdelava aplikacije	7
Visual Studio	7
Xamarin	8
4.REZULTATI	25
4.1 Izdelana aplikacija	25
4.2 Rezultati testiranja	29
5. RAZPRAVA	31
5.1 Hipoteze	31
6. ZAKLJUČEK	32
7. POVZETEK	33
8. ZAHVALE	34
VIRI IN LITERATURA	35

KAZALO SLIK

Slika 1: Google Zemljevidi (vir: Geo awesomeness, [13])	5
Slika 2: Apple Zemljevidi (vir: Apple Support, [14])	6
Slika 3: IDE Visual Studio (vir: code.VisualStudio.com)	8
Slika 4: Vtičnik Xamarin (vir: MS Power User, [15])	9
Slika 5: Vtičniki, uporabljeni v projektu (vir: lastni)	10
Slika 6 : Zahteva lokacije oz. koordinat (vir: lastni)	11
Slika 7: Prikaz pina na zemljevidu (vir: lastni)	11
Slika 8 : Risanje črt na zemljevidu (vir: lastni)	12
Slika 9: Izračun razdalje med uporabnikom in izbrano točko (vir: lastni)	12
Slika 10: Prikaz poti od avtobusne postaje do Šolskega centra (vir: lastni)	13
Slika 11: Razredi za podatke v podatkovni bazi (vir: lastni)	14
Slika 12: Koda za dodajanje poti na zemljevid (vir: lastni)	15
Slika 13: Preverjanje, če je uporabnik blizu točke (vir: lastni)	16
Slika 14: Predvajanje zvoka "Niste na pravem mestu" (vir: lastni)	16
Slika 15: Prikaz poti okoli Škalskega jezera (vir: lastni)	17
Slika 16: Prikaz poti po Sončnem parku (vir: lastni)	18
Slika 17: Izbira poti (vir: lastni)	19
Slika 18: Izbira poti – Pot ob cesti (vir: lastni)	20
Slika 19: Izbira poti – Pot skozi naselje (vir: lastni)	21
Slika 20: Pot med Šolskim centrom Velenje in Sončnim parkom (vir: lastni)	22
Slika 21: Razredi za podatke iz formata JSON (vir: lastni)	23
Slika 22: Izbira točke, do katere aplikacija glasovno usmerja (vir: lastni)	24
Slika 23: Prikaz znamenitosti na zemljevidu (vir: lastni)	26
Slika 24: Stran Znamenitosti (vir: lastni)	27
Slika 25: Stran Poti (vir: lastni)	28

1. UVOD

Dandanes ima vsak pametni telefon možnost nameščanja in uporabe aplikacij oziroma programov, ki so napisani posebej za mobilne naprave. Med temi najdemo tudi aplikacije s področja turizma. Želeli smo združiti ta dva pojma in izdelati osebnega turističnega vodiča, ki ga lahko imaš v lastnem žepu.

V okviru raziskovalne naloge smo želeli izdelati aplikacijo, s katero bi turistom omogočili ogled mesta in znamenitosti med hojo in kolesarjenjem s čim manj motnjami. To smo hoteli doseči z zvočnim vodenjem v aplikaciji, da bi turiste njihov telefon čim manj motil, ker bi morali pogosto gledati na njegov zaslon.

Za izdelavo našega projekta je bila glavna motivacija enostavnejši in zanimivejši prikaz znamenitosti v Velenju. S tem bi omogočili prijetnejšo izkušnjo še drugim prebivalcem Slovenije in turistom, ki obiskujejo Velenje. Želeli smo jim omogočiti, da jim poleg tega, da jih aplikacija vodi po poteh v Velenju in okolici, pove tudi kakšno zanimivost in tako razširi njihovo znanje.

Dela smo se lotili tako, da smo se najprej naučili, kako programirati v vtičniku Xamarin za Visual Studio, potem pa smo v aplikaciji začeli ustvarjati poti, katerim smo kasneje dodali tudi glasovno vodenje.

1.1 Hipoteze

Pred začetkom razvoja aplikacije smo si zadali naslednji hipotezi:

1. Mobilna aplikacija z glasovnim vodenjem uporabniku olajša navigiranje med potjo.
2. Mobilna aplikacija z glasovnim vodenjem je pri uporabi natančna na 10 metrov.

2. PREGLED STANJA TEHNIKE

2.1 Mobilne aplikacije

Mobilna aplikacija ali tudi samo aplikacija, je program, ki je posebno napisan za mobilne naprave, kot so tablice in pametni telefoni ter dandanes tudi pametne ure. Najprej so bile aplikacije namenjene za bolj produktivne namene, na primer klicanje, pisanje elektronski sporočil, pisanje sporočil, delanje zapiskov itd., dandanes pa veliko podjetij dela aplikacije, ki so namenjene predvsem razvedrilu. [1][2]

Za razvijanje mobilnih aplikacij se uporabljajo posebna programska orodja (angl. Integrated Development Environment oz. IDE), ki so namenjena razvijanju za mobilne naprave, kot so npr. Android Studio (za Android aplikacije) in xCode (za iOS aplikacije). Nekatera orodja, kot npr. Xamarin ali Apache Cordova, omogočajo tudi razvijanje aplikacij za več platform naenkrat (angl. Cross-platform). Aplikacije lahko testiramo tudi na osebnih računalnikih, saj večina IDE okolij ponuja emulatorje, to so programi, ki simulirajo neko drugo napravo na računalniku (v našem primeru telefon ali tablico) in so že vgrajeni v sam IDE. [3]

Te aplikacije se lahko prenašajo in nameščajo na tako imenovanih "App Store" oziroma trgovinah. Najbolj znani so Google Play, App Store in Microsoft Store. Platforma Android uporablja Google Play. Tam lahko razvijalci vseh kalibrov prodajajo svoje aplikacije, čeprav je samo okoli 20 % aplikacij na Google Play plačljivih, saj jih večina služi z reklamami. App Store je razvilo podjetje Apple za svojo platformo iOS. Večina aplikacij na App Store od svojega začetka ni dobila nobenega prenosa, zato jih nekateri tudi imenujejo "Zombie apps". [4]

Microsoft Store je Microsoftova rešitev za distribucijo aplikacij na njihovi lastni platformi, Windows Phone. Čeprav se tam tudi znajde kakšna namizna aplikacija, se večinoma uporablja za mobilne naprave. [5]

Pri razvijanju mobilnih aplikacij moramo biti pazljivi na ciljno platformo in njene funkcije. Mobilne naprave imajo različno programsko in strojno opremo, kar zelo vpliva na združljivost z aplikacijami. Pomemben je tudi uporabniški vmesnik, saj je to prva stvar, ki jo vidimo, ko zaženemo aplikacijo.

2.2 GPS

Globalni sistem pozicioniranja (angl. Global Positioning System oz. GPS) je navigacijski sistem, ki uporablja satelite za določanja lege določene točke kjerkoli na Zemlji ali na zemeljski tirnici. GPS sistem je zasnovalo in ga tudi trenutno upravlja obrambno ministrstvo ZDA. Cena tega vzdrževanja je okoli 400 milijonov dolarjev na leto. Sistem lahko vsak brezplačno uporablja, če ima primeren sprejemnik, ki pa je vgrajen praktično v vsak pametni telefon.

Za delovanje tega sistema okoli Zemlje potuje 24 satelitov, ki konstantno pošiljajo čas (svoje atomske ure) in podatke gibanja. Za določanje pozicije sprejemnika je potrebna zelo velika natančnost ur v satelitih. Ti sateliti Zemljo dnevno obkrožijo dvakrat, letijo pa 20200 km nad Zemljo. Za pridobitev koordinat določenega sprejemnika so potrebni 4 sateliti. Za izračun so potrebne tudi njihove pozicije, ki jih dobimo iz poslanih signalov. Sprejemnik se nahaja na središču posameznih sfer teh satelitov. Teoretično so potrebni samo trije sateliti, saj bi isti izračun dobil podobne rezultate s presečiščem samo treh sfer.

V pametnih telefonih so vgrajeni GPS sprejemniki, ki so potrebni za računanje GPS koordinat naprave. Najpogosteje se GPS sprejemniki v telefonih uporabljajo za navigacijo ter za aplikacije, ki so namenjene za iskanje izgubljenih telefonov. GPS se uporablja tudi za izdelovanje metapodatkov, to so podatki o neki datoteki, ki niso vključeni vanjo (npr. dimenzije slike, kraj izdelave ...).

GPS se vedno pogosteje uporablja tudi v mobilnih igrah, ki vstavijo virtualne 3D objekte v resnični svet s pomočjo razširjene resničnosti (angl. Augmented Reality oz. AR). Določena socialna omrežja, kot sta npr. Instagram in Facebook, prav tako uporabljajo GPS, in s tem omogočajo uporabnikom, da ob objavljeni sliki tudi pokažejo, kje so bili. Nekatera socialna omrežja, kot je npr. Snapchat, pa imajo na voljo aktiven zemljevid, s katerim lahko vidiš, kje se trenutno nahajajo tvoji prijatelji.

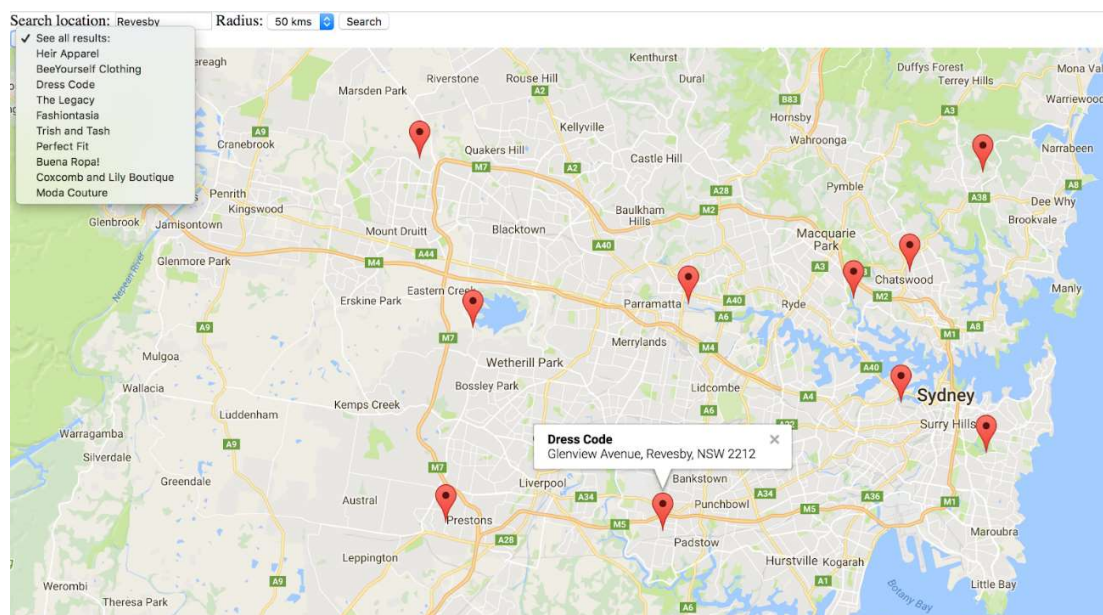
Ena izmed najbolj znanih GPS aplikacij je Google Maps, ki jo vsebuje vsak Android telefon in prikazuje uporabniku najbolj optimalne poti, po katerih se lahko vozi oziroma sprehaja. Google je razvil tudi programski vmesnik (angl. Application Programmable Interface oz. API) z imenom Google Maps API, ki omogoča uporabo njihovega zemljevida v projektih in aplikacijah. [6] [2]

Programska orodja IDE, kot sta npr. Xamarin ali Android Studio, imajo vgrajene tudi svoje vtičnike za uporabo GPS sistema. Z njimi lahko dostopamo do sprejemnika v naši napravi in dobimo koordinate naprave.

2.3 Podobne aplikacije

Google Zemljevidi

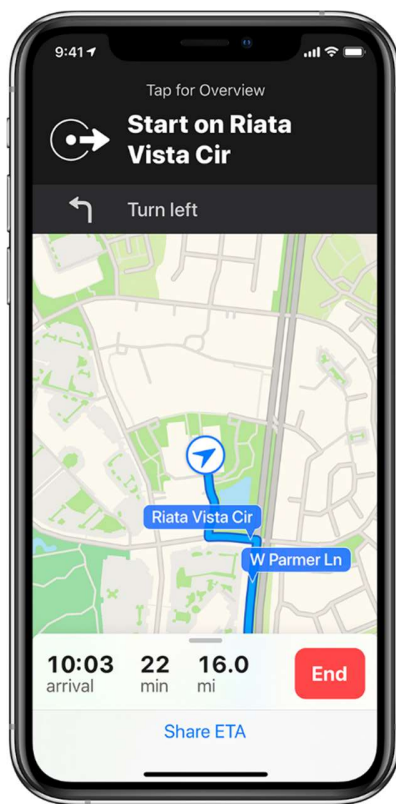
Je ena od najbolj znanih aplikacij za navigiranje po cestah. Vsebuje tudi glasovno vodenje, ki deluje na osnovi tehnologije Text-To-Speech, ki samo prebere mesto, kjer mora uporabnik zaviti. Aplikacija Google Zemljevidi vodi po poteh, primernih za motorna vozila ali pešce ter ima tudi možnosti navigiranja z javnim prevozom. Uporabnika vodi po najkrajši poti in mu sproti nudi tudi alternativne poti, ki so iste dolžine ali pa tudi daljše. Uporabniku lahko prikaže tudi najbližje restavracije, hotele in podobno ter tudi vodi do njih, če uporabnik to želi. Uporabnik lahko na zemljevidu tudi shrani poljubne lokacije, kot sta na primer dom in služba. Google Zemljevidi je tudi primarna aplikacija za navigacijo v sistemu Android. [7][8]



Slika 1: Google Zemljevidi (vir: Geo awesomeness, [13]).

Apple Maps

Apple Maps je primarna aplikacija za navigacijo v sistemu iOS. Je zelo podobna Google Zemljevidom, čeprav so se uporabniki pritoževali nad njeno uporabo, saj ni bila zelo natančna in ni vodila po pravih poteh. Vsebuje dodatne možnosti, kot so notranja zgradba večjih stavb (npr. letališča in nakupovalni centri). Aplikacija je bila zelo izboljšana od svojih začetkov, na iOS sistemu pa je poleg nje še vedno zelo priljubljena tudi aplikacija Google Zemljevidi. Vsebuje tudi glasovno navigacijo, ki je zelo podobna rešitvi podjetja Google. [9]



Slika 2: Apple Zemljevidi (vir: Apple Support, [14]).

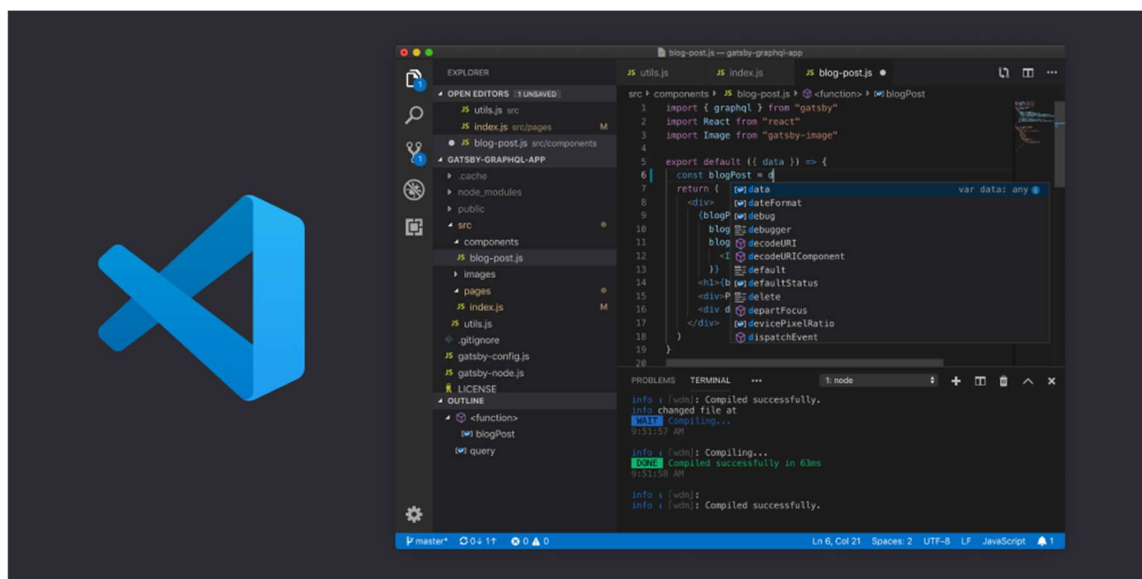
3.METODOLOGIJA

3.1 Izdelava aplikacije

Izdelali smo mobilno aplikacijo, ki uporabniku omogoča vodenje po več vnaprej določenih poteh. Za razliko od drugih navigacijskih aplikacij je naša narejena bolj natančno samo za Velenje, to pomeni, da med vodenjem po poteh opisuje tudi znamenitosti, ki se nahajajo na poti. Poleg tega naša aplikacija na zemljevidu prikaže tudi pot od uporabnikove trenutne lokacije do znamenitosti, ki si jo uporabnik lahko izbere na zemljevidu.

Visual Studio

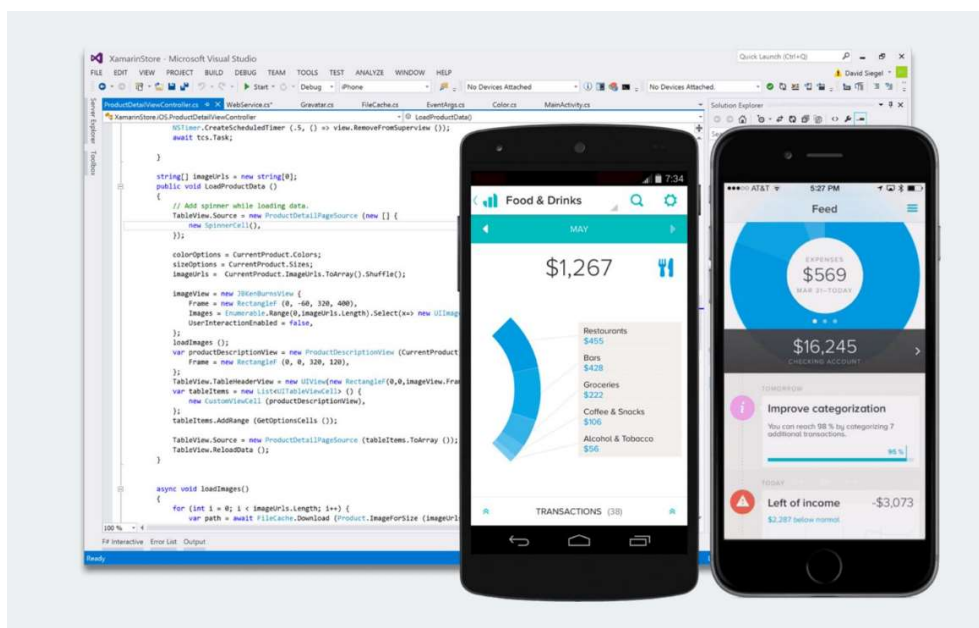
Za razvoj naše mobilne aplikacije smo uporabili IDE orodje Visual Studio. Ustvarilo ga je podjetje Microsoft in je namenjeno za razvijanje programov za operacijske sisteme Windows, spletne strani, spletne aplikacije, mobilne aplikacije itd. Za to uporablja različne razvojne platforme, kot so Windows Forms in Windows API. Čeprav je okolje namenjeno omejenemu številu operacijskih sistemov, se vseeno uporablja za razvoj aplikacij za mobilne telefone in spletne strani. To je mogoče preko vtičnikov, kot je npr. Xamarin. [10]



Slika 3: IDE Visual Studio (vir: code.VisualStudio.com).

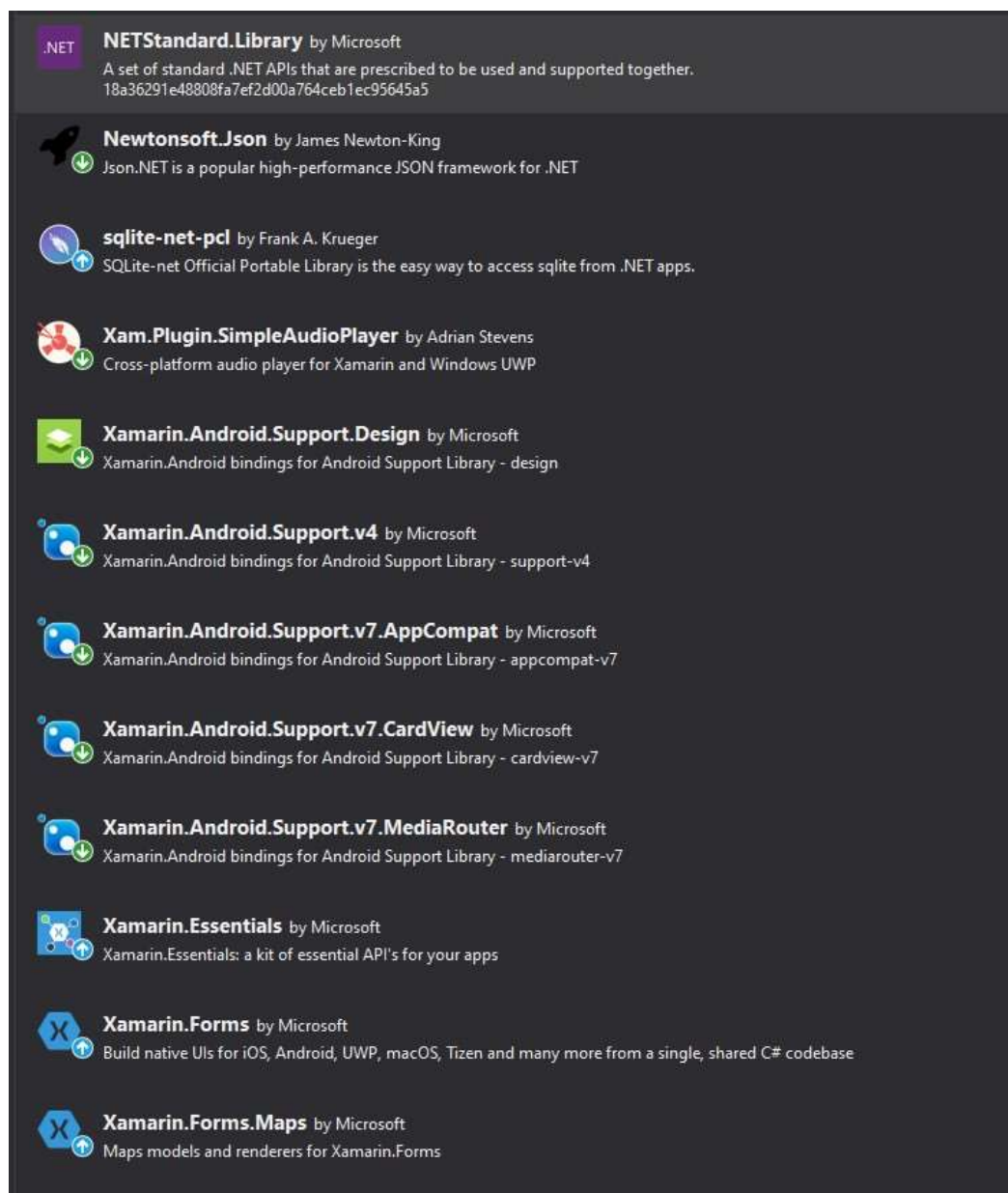
Xamarin

Xamarin je vtičnik za Visual Studio, ki omogoča razvoj mobilnih aplikacij v prej omenjenem IDE. Omogoča razvoj za Android (s komponento Xamarin.Android) in za iOS (s Xamarin.iOS). Za programiranje izgleda aplikacije se uporablja XAML, ki je programski jezik za pozicioniranje gradnikov, kot so npr. gumbi ali pa besedilo, v uporabniški vmesnik aplikacije. Za programiranje logike aplikacije se uporablja programski jezik C#, ki je nadgradnja C++. Aplikacija je narejena s pomočjo Google Maps API, ki omogoča, da v svoji aplikaciji uporabiš zemljevid, ki ga je naredil Google, z njim prikazuješ poti ter računaš dolžine na tem zemljevidu. [11]



Slika 4: Vtičnik Xamarin (vir: MS Power User, [15]).

Za aplikacijo smo uporabili knjižnice in vtičnike Xamarin.Forms, Xamarin.Form.Maps, Xamarin.Essentials, Xam.Plugin.SimpleAudioPlayer, sqlite-net-pcl in Newtonsoft.Json.



Slika 5: Vtičniki, uporabljeni v projektu (vir: lastni).

Najprej smo za preverjanje lokacije ustvarili gumb v aplikacij, ob kliku nanj se prikažejo naše trenutne koordinate. Te koordinate dobimo z ukazoma na Sliki 6.

```
var request = new GeolocationRequest(GeolocationAccuracy.Best);  
var location = await Geolocation.GetLocationAsync(request);
```

Slika 6 : Zahteva lokacije oz. koordinat (vir: lastni).

Te podatke shranimo in jih lahko izpisujemo z `location.Latitude` (lega severno oz. južno od ekvatorja) in `location.Longitude` (lega zahodno oz. vzhodno od primarnega - Greenwiškega poldnevnika).

Potem smo v našo aplikacijo dodali Google zemljevid z uporabo Google Maps API. Tega smo dobili tako, da smo ustvarili račun na platformi za storitve v oblaku Google Cloud Platform, kjer smo dobili zastoj API ključ. Ko smo dodali zemljevid, smo dodali še en gumb, ki nam ob kliku odpre ta zemljevid. Če kliknemo kamorkoli na zemljevidu, nam bo na to lokacijo postavilo oznako (angl. pin) ter narisalo ravno črto do tja in nam napisalo, koliko kilometrov smo oddaljeni od izbrane lokacije.

Za risanje oznak oz. pinov moramo le-te najprej ustvariti in jim dodati lastnosti (oznako, naslov, tip pina in lokacijo), potem pa ga dodamo na zemljevid (Slika 7).

```
Pin pin = new Pin  
{  
    Label = "Avtobusna postaja Velenje",  
    Address = "Agencija za prodajo avtobusnih vozovnic",  
    Type = PinType.Place,  
    Position = new Position(46.358679, 15.118278)  
};  
mapdemo.Pins.Add(pin);
```

Slika 7: Prikaz pina na zemljevidu (vir: lastni).

Podobno naredimo tudi s črtami, ki označujejo poti na zemljevidu (angl. polyline), ki jim ob ustvarjanju dodamo lastnosti kot so: barva, širina, začetna in končna lokacija ter jih dodamo na zemljevid (Slika 8).

```
Polyline polyline = new Polyline
{
    StrokeColor = Color.Blue,
    StrokeWidth = 12,
    Geopath =
    {
        new Position(location.Latitude, location.Longitude),
        new Position(App.latitude, App.longitude)
    }
};

maap.MapElements.Add(polyline);
```

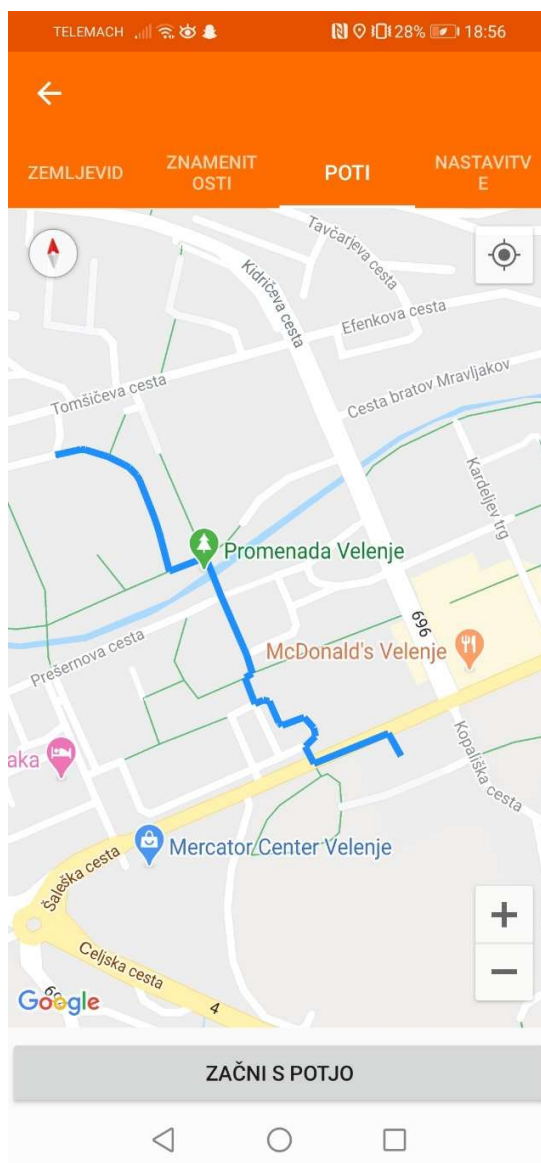
Slika 8 : Risanje črt na zemljevidu (vir: lastni).

Za oddaljenost od izbrane točke pa uporabimo ukaz CalculateDistance (Slika 9).

```
double distance = Location.CalculateDistance(location, lok, DistanceUnits.Kilometers);
label2.Text = $"Med tabo in izbrano točko je: {distance}";
```

Slika 9: Izračun razdalje med uporabnikom in izbrano točko (vir: lastni).

Glavni del aplikacije oziroma glasovno vodenje smo naredili na novi strani (angl. page), kjer lahko uporabnik izbere eno izmed vnaprej programiranih poti, po kateri si želi glasovno vodenje. Najprej smo izdelali pot od avtobusne postaje do območja Šolskega centra Velenje (Slika 10), saj je ta pot krajša in je lažja za testiranje. Najprej smo s polyline ročno narisali pot. Potem smo si na poti izbrali določene lokacije, kjer želimo, da se predvajajo zvočni posnetki.



Slika 10: Prikaz poti od avtobusne postaje do Šolskega centra (vir: lastni).

Po določitvi točk na poti smo v aplikacijo dodali tudi podatkovno bazo sqlite, kamor smo shranili vse podatke za izdelavo črt in risanje pinov. V bazi imamo shranjene vse poti in vse koordinate za vnaprej določene poti, za katere smo naredili tudi razrede (Slika 11).

```

namespace App1
{
    [Table("koordinata")]
    public class koordinate
    {
        [PrimaryKey, AutoIncrement]
        public int id { get; set; }
        public double latitude { get; set; }
        public double longitude { get; set; }
        public int pot_id { get; set; }
    }

    [Table("poti")]
    public class poti
    {
        [PrimaryKey, AutoIncrement]
        0 references
        public int id { get; set; }
        0 references
        public string ime { get; set; }
    }

    [Table("tocke")]
    19 references
    public class tocke
    {
        [PrimaryKey, AutoIncrement]
        0 references
        public int id { get; set; }
        5 references
        public string ime { get; set; }
        5 references
        public string opis { get; set; }
        0 references
        public int pot_id { get; set; }
        10 references
        public double latitude { get; set; }
        10 references
        public double longitude { get; set; }
        0 references
        public int st_tocke { get; set; }
        5 references
        public string audio { get; set; }
    }

    [Table("zanimivosti")]
    10 references
    public class zanimivost
    {
        [PrimaryKey, AutoIncrement]
        0 references
        public int id { get; set; }
        2 references
        public string ime { get; set; }
        2 references
        public string opis { get; set; }
        4 references
        public double latitude { get; set; }
        4 references
        public double longitude { get; set; }
    }
}

```

Slika 11: Razredi za podatke v podatkovni bazi (vir: lastni).

Podatkovno bazo smo nato povezali s projektom in napisali kodo, ki riše poti s podatki iz baze (Slika 12).

```
1 reference
private async void Narisi_pot()
{
    Color barva = (Color)(converter.ConvertFromInvariantString(App.color));
    podatki = await db.GetkoordinatapotAsync(pot_id);

    for (int i = 0; i < (podatki.Count - 1); i++)
    {
        koordinata a = podatki[i];
        double longic = a.longtitude;
        double lant = a.latitude;

        int o = i + 1;
        koordinata b = podatki[o];
        double longic2 = b.longtitude;
        double lant2 = b.latitude;

        Polyline polyline = new Polyline
        {
            StrokeColor = barva,
            StrokeWidth = 12,
            Geopath =
            {
                new Position(lant,longic),
                new Position(lant2,longic2)
            }
        };
        mapdemo.MapElements.Add(polyline);
    }
}
```

Slika 12: Koda za dodajanje poti na zemljevid (vir: lastni).

Za preverjanje, kdaj se uporabnik približa točki na poti, aplikacija računa razdaljo med uporabnikom in naslednjo točko na poti. Ko se ji približa za 10 metrov ali manj, se predvaja zvok (Slika 13).

```
while (na_tocki == false)
{
    var request = new GeolocationRequest(GeolocationAccuracy.Best);
    var location = await Geolocation.GetLocationAsync(request);
    double distance = Location.CalculateDistance(location, lok, DistanceUnits.Kilometers);

    if (distance < 0.01)
    {
        player.Load(tocki.audio);
        player.Play();

        na_tocki = true;
    }
}
```

Slika 13: Preverjanje, če je uporabnik blizu točke (vir: lastni).

Zvoke smo v aplikaciji predvajali z vtičnikom SimpleAudioPlayer. Ker želimo, da uporabniki točke obiskujejo po vrsti in ne preskakujejo poti, smo v podatkovni bazi označili vrstni red točk, in ko uporabnik pride v bližino ene izmed točk, se predvaja posnetek za to točko in na zemljevidu s pinom označi naslednjo točko. To mu omogoči, da točke obiše po vrsti. Če uporabnik želi začeti pot, mora najprej stati na začetni točki poti in pritisniti na gumb začetek, v nasprotnem primeru pa se predvaja zvok, ki pove, da ni na pravem mestu (Slika 14).

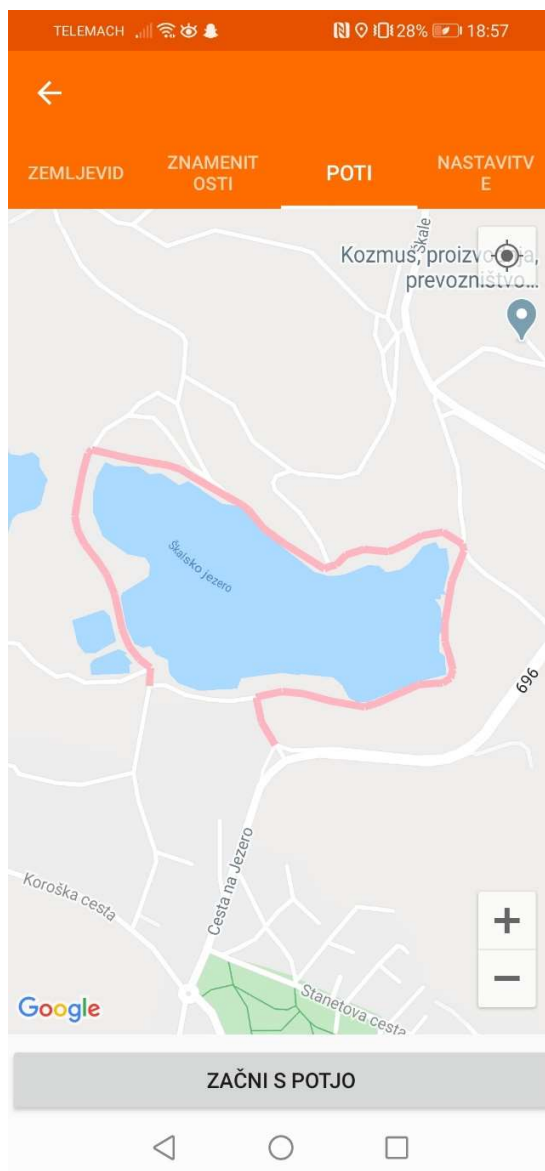
```
if (played == false)
{
    player.Load("not.mp3");
    player.Play();

    played = true;
}
```

Slika 14: Predvajanje zvoka "Niste na pravem mestu" (vir: lastni).

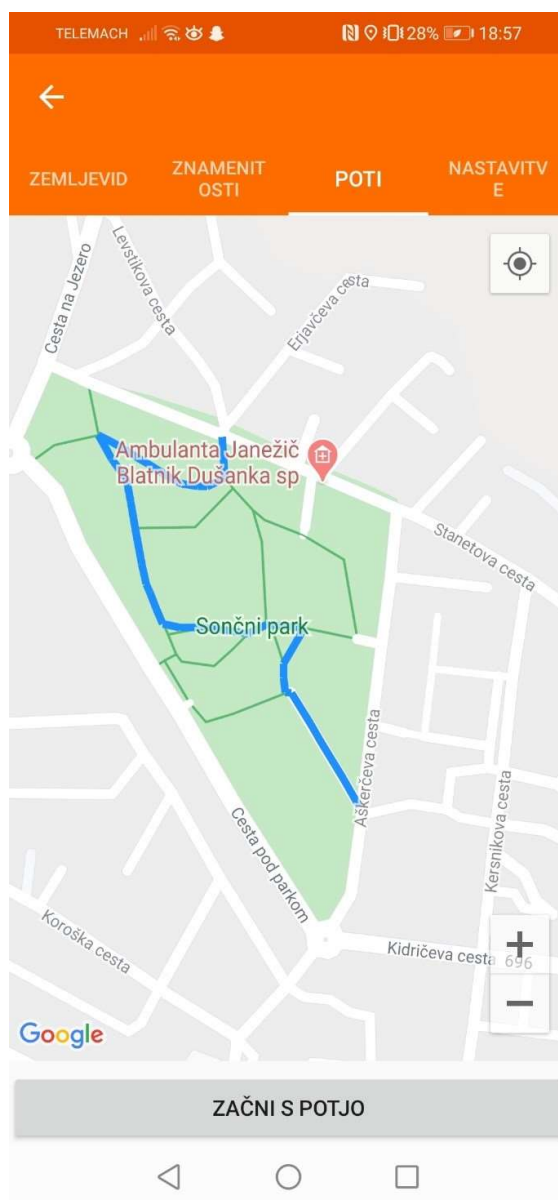
Za drugo pot smo si izbrali pot okoli Škalskega jezera (Slika 15). Pri tej poti smo porabili veliko več polyline, saj ima pot veliko ovinkov. Ostalo smo naredili na isti način kot pri poti

od avtobusne postaje do Šolskega centra, a smo si vzeli več časa in se bolj potrudili, saj je pot okoli jezera turistično bolj zanimiva kot pa prva pot, ki smo jo izdelali predvsem za testiranje aplikacije.



Slika 15: Prikaz poti okoli Škalskega jezera (vir: lastni).

Tudi tretjo pot, ki vodi po Sončnem parku, smo opremili z opisi znamenitosti (Slika 16).

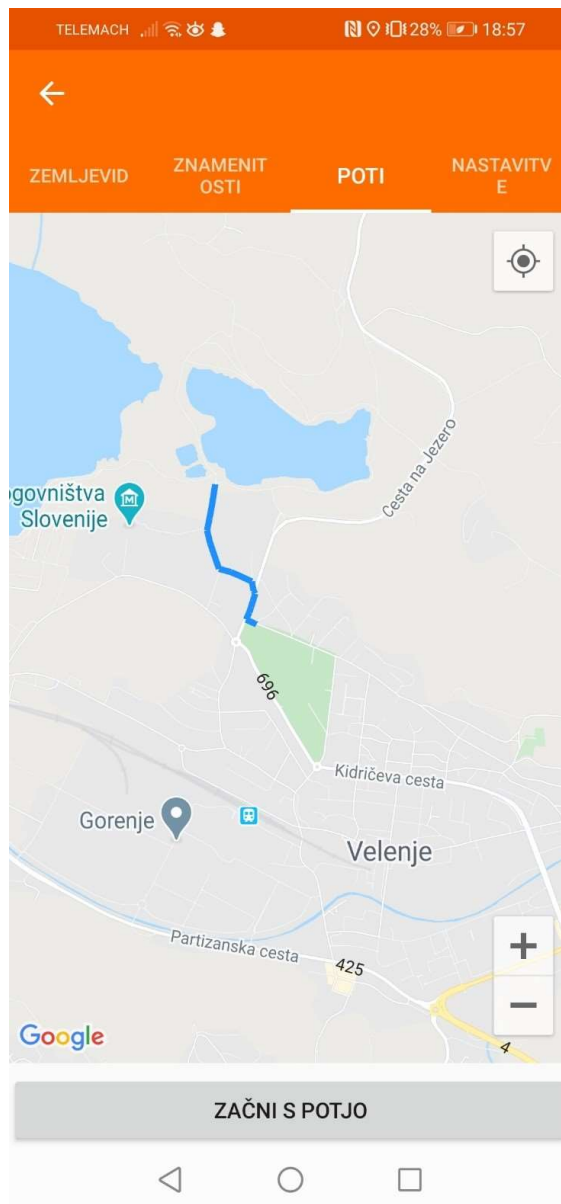


Slika 16: Prikaz poti po Sončnem parku (vir: lastni).

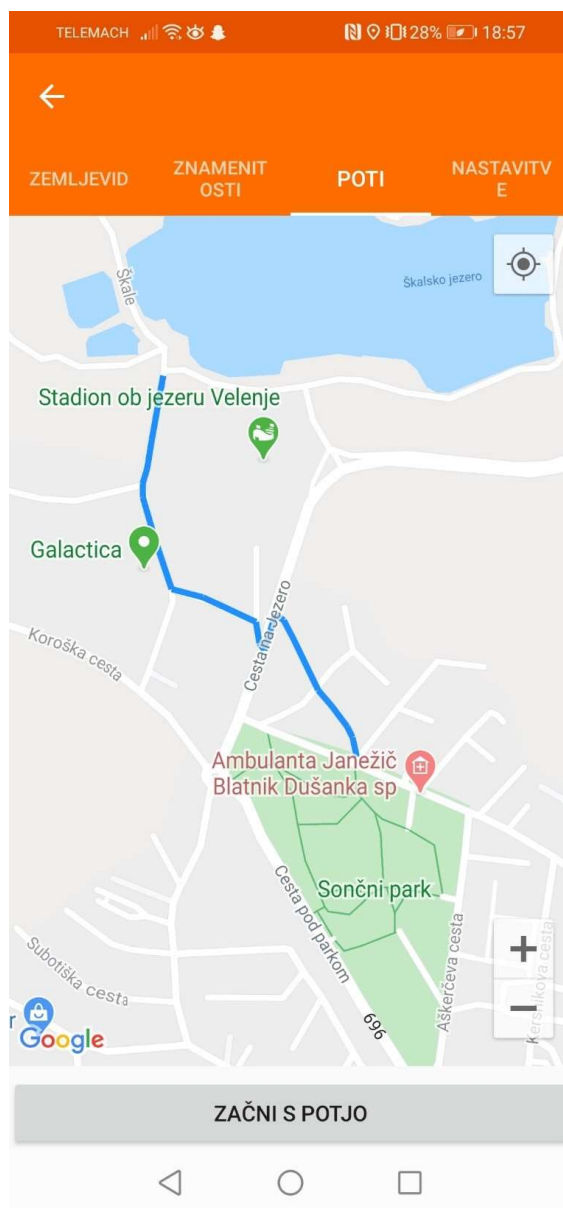
Pri četrti poti, ki vodi med Sončnim parkom in Škalskim jezerom, smo poskusili nekaj novega. Uporabnik lahko izbere, po kateri cesti želi, da ga pot vodi. To smo naredili s pomočjo okenca, ki uporabniku ponudi dve izbiri (Slika 17).



Slika 17: Izbira poti (vir: lastni).

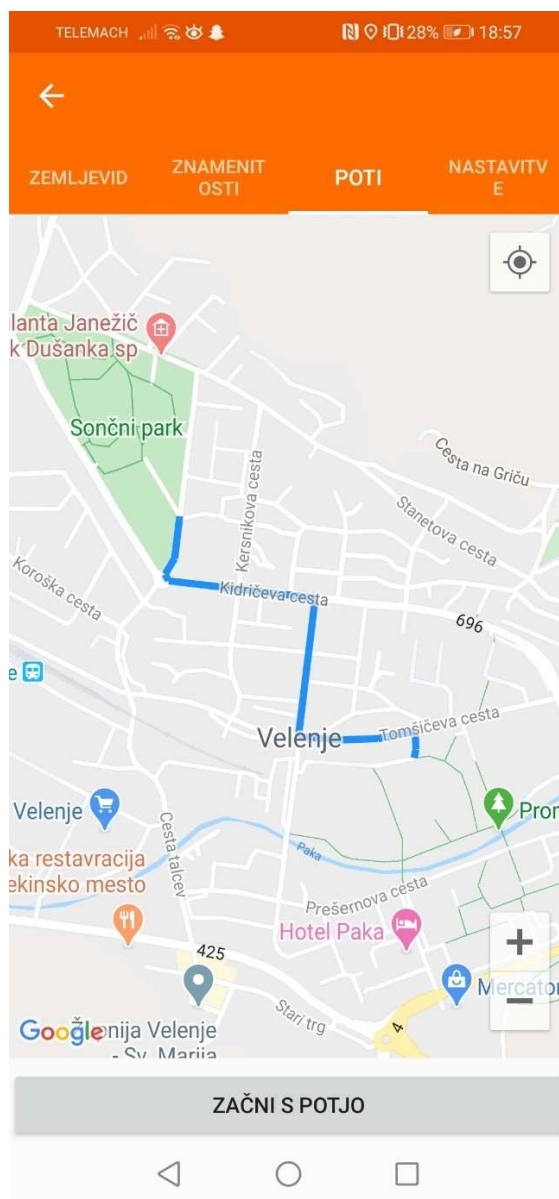


Slika 18: Izbira poti – Pot ob cesti (vir: lastni).



Slika 19: Izbira poti – Pot skozi naselje (vir: lastni).

Naslednjo pot, ki vodi od Šolskega centra Velenje do Sončnega parka (Slika 20), smo dodali z namenom, da povežemo vse poti med sabo. Na tej poti ni kakšnih znamenitosti, pač pa aplikacija uporabnika samo glasovno vodi po pravi poti.



Slika 20: Pot med Šolskim centrom Velenje in Sončnim parkom (vir: lastni).

Ko smo končali s potmi, smo v aplikaciji uporabili storitev Google Directions API, ki samodejno po cestah nariše pot med dvema točkama. To smo dosegli tako, da aplikacija storitvi pošlje določene parametre (npr. začetna točka, končna točka, API ključ ...) in dobi nazaj podatke v formatu JSON. Narediti smo morali posebne razrede za vse vrnjene podatke

(Slika 21), nato pa smo uporabili maneuver (angl. Maneuver), ki ga aplikacija dobi z vsakim ovinkom (npr. turn-left). Naredili smo glasovne posnetke z istimi imeni kot manevri, ki smo jih nato predvajali na vsakem ovinku.

```
1reference
public class Route
{
    0 references
    public Bounds bounds { get; set; }
    0 references
    public string copyrights { get; set; }
    9 references
    public List<Leg> legs { get; set; }
    2 references
    public OverviewPolyline overview_polyline { get; set; }
    0 references
    public string summary { get; set; }
    0 references
    public List<object> warnings { get; set; }
    0 references
    public List<object> waypoint_order { get; set; }
}

6 references
public class RootObject
{
    0 references
    public List<GeocodedWaypoint> geocoded_waypoints { get; set; }
    11 references
    public List<Route> routes { get; set; }
    0 references
    public string status { get; set; }
}
```

Slika 21: Razredi za podatke iz formata JSON (vir: lastni).

Uporabnik lahko v aplikaciji na straneh Zemljevid in Znamenitosti izbere določeno točko, aplikacija pa ga nato na ta način vodi do nje (Slika 22).

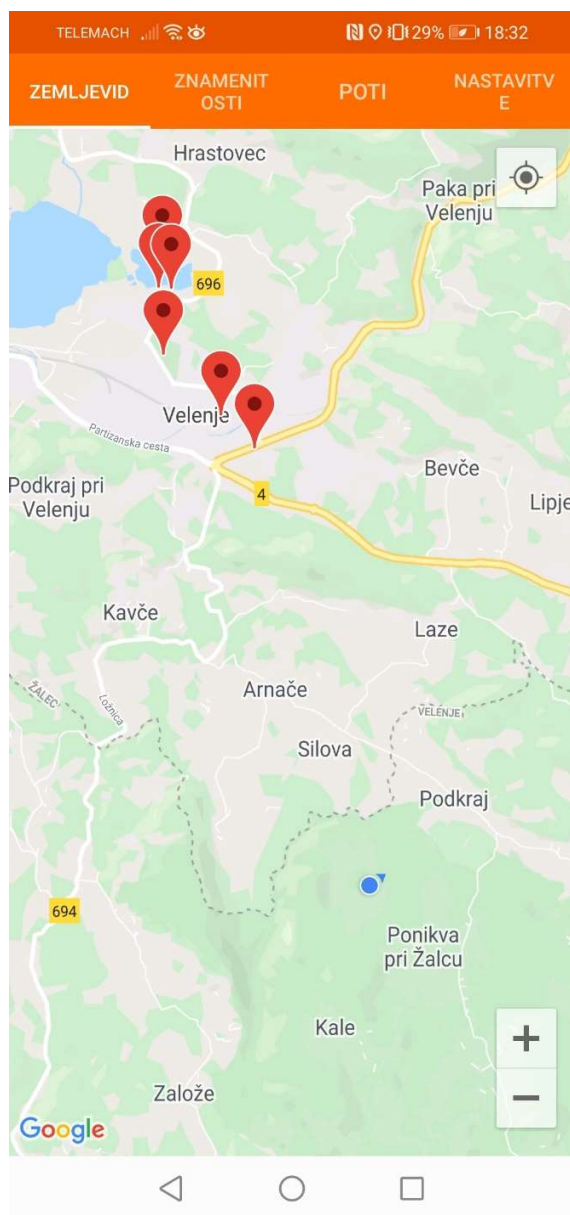


Slika 22: Izbira točke, do katere aplikacija glasovno usmerja (vir: lastni).

4. REZULTATI

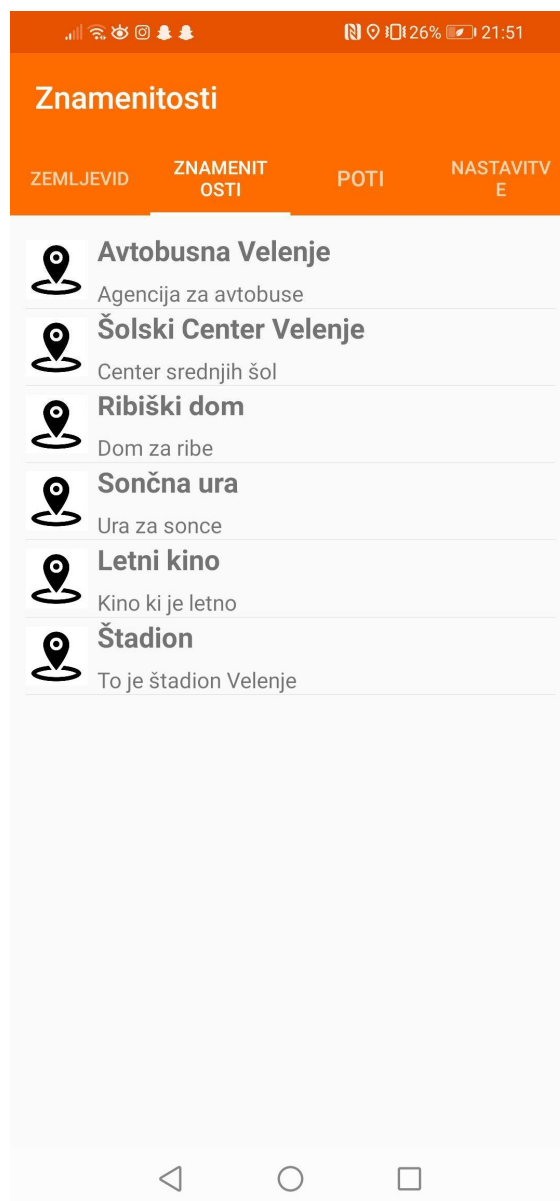
4.1 Izdelana aplikacija

Ko aplikacijo zaženemo, se nam najprej pokaže stran z zemljevidom, na katerem so že prikazane vse znamenitosti (Slika 23). Če kliknemo na znamenitost, se nam prikaže pot do te točke, nato pa nas aplikacija glasovno vodi to nje.



Slika 23: Prikaz znamenitosti na zemljevidu (vir: lastni).

Na drugi strani so izpisane vse znamenitosti (Slika 24), in ko kliknemo nanjo, se prikaže zemljevid s potjo, aplikacija pa uporabnika glasovno vodi do nje.



Slika 24: Stran Znamenitosti (vir: lastni).

Na strani Poti so prikazane vse naprej določene poti - do zdaj jih imamo 5 (Slika 25). Ko kliknemo na pot, se nam prikaže zemljevid, na katerem so prikazane točke, kjer ima vsaka svoj glasovni posnetek.



Slika 25: Stran Poti (vir: lastni).

Pot avtobusne postaje do Šolskega centra Velenje in nato do Sončega parka sta manj opremljeni in sta glasovno vodeni s tehnologijo Text-To-Speech. Ena pot uporabnika vodi

skozi podhod in po najkrajši poti do Šolskega centra. Ta pot je bila ustvarjena kot prikaz delovanja, da smo ugotovili, kako delujejo zemljevidi v Xamarinu. Pot do Sončnega parka pa povezuje z ostalimi potmi.

Pot okoli Škalskega jezera ima boljše glasovno vodenje (posnet glas, kateri nam ob aktivaciji pove nekaj o trenutnemu okolju oziroma točki). Ta pot je odlična za sprehod ali kolesarjenje okoli Škalskega jezera, saj lahko prostoročno izvemo nekaj o tistem okolju in točkah.

Pot po Sončnega parka je prav tako kar dobro opremljena, saj je na njej in v samem parku veliko zanimivosti.

Pot od parka do jezera ima dve možnosti: pot čez naselje ali pot ob glavni cesti. Ta pot je dobro opremljena z glasovnim vodenjem po cestah in je primerna za kolesarjenje ter sprehode.

4. 2 Rezultati testiranja

Ker smo aplikacijo vseskozi nadgrajevali, smo testiranje velikokrat ponovili.

Najprej smo testirali pot po Sončnem parku, kjer so se pojavili prvi problemi. Točk nam ni prikazovalo po vrsti, zato smo spremenili način preverjanja ali je uporabnik že na točki. Prej smo namreč preverjali ali je uporabnik v vnaprej določenem kvadratu okoli točke, nato pa smo začeli s preverjanjem oddaljenosti do točke, kar je rešilo to težavo.

Nato smo testirali še ostale poti, kjer pa problemov nismo imeli.

Po testiranju poti smo začeli s testiranjem samodejnega risanja poti med dvema točkama. Zaradi pandemije smo morali testirati preko Xamarin emulatorja, kjer smo lahko narisali poti, kjer smo želeli, da se navidezno premikamo. Za to smo porabili več časa, vendar smo z več testiranjem in popraviljem napak probleme hitro odpravili.

Na koncu smo testirali še, če se predvajajo posnetki za glasovno vodenje na samodejno izrisani poti. To nam najprej ni delalo, a smo napake hitro popravili.

Aplikacija sedaj deluje brezhibno.

5. RAZPRAVA

Glede na čas, v katerem smo naredili aplikacijo, se nam zdi aplikacija zelo dobra, saj smo uspeli težave, ki so se pojavile, odpraviti, in sedaj deluje odlično.

Aplikaciji sicer manjka lepši izgled in še več poti, za kar pa bi potrebovali več časa, kot smo ga imeli na voljo.

Aplikacija se lahko primerja z Google Zemljevidi, saj ima podobne funkcije in dodano funkcionalnost prostoročnega vodenja. Razlika med njima je predvsem v tem, da je naša aplikacija bolj natančna in poleg vodenja, kje in kam moramo zaviti, predvaja še nekaj zanimivih informacij o sami poti in morebitnih znamenitostih ob njej. Vendar je trenutno omejena samo na poti, ki smo jih ustvarili.

5.1 Hipoteze

Hipotezo 1, da mobilna aplikacija z glasovnim vodenjem uporabniku olajša navigiranje med potjo, lahko potrdimo, saj smo aplikacijo nekajkrat testirali in smo vedno lahko prišli do cilja, ne da bi se ozrli na pametni telefon.

Hipotezo 2, da bo naša mobilna aplikacija z glasovnim vodenjem natančna na 10 metrov, lahko potrdimo, saj smo tekom testiranja videli, da se nam pravočasno začnejo predvajati zvoki na izbranih lokacijah. Aplikacija namreč računa, koliko je uporabnik oddaljen od točke, in če je oddaljenost manjša kot 10 metrov, se predvaja glasovni posnetek.

6. ZAKLJUČEK

S to raziskovalno nalogo smo nadgradili svoje znanje programskega jezika C# IDE okolja Visual Studio ter spoznali razvoj mobilnih aplikacij z uporabo vtičnika Xamarin. Okrepilo se je tudi naše znanje o lokacijskih storitvah ter kako jih lahko uporabimo v naših programih.

Zelo nam je bilo všeč delo z vtičnikom Xamarin, saj ni bilo preveč zahtevno, to znanje pa nam bo v prihodnosti koristilo. Xamarin Forms dokumentacija, s katero smo si pomagali, ko smo naleteli na probleme, je pregledna in enostavna za uporabo.

V aplikaciji smo uporabili tudi bazo podatkov, naredili več poti, samodejno vodenje ter izdelali uporabniški vmesnik. Za objavo aplikacije v trgovinah Google Play ali Apple App Store pa bi potrebovali še več poti, boljše glasovne posnetke in mogoče tudi možnost izbire med moškim in ženskim glasom.

Menimo, da bodo takšne in podobne aplikacije nadomestile turistične vodiče, saj bomo lahko z razvojem tehnologije obogatene resničnosti (angl. Augmented Reality) bolje prikazali zanimivosti.

7. POVZETEK

Zastavili smo si problem, da pomagamo narediti Velenje bolj dostopno in bolj zanimivo mesto za turiste, ki si mesto ogledujejo peš ali s kolesom in jih zanimajo različne teme. To smo dosegli z mobilno aplikacijo. Aplikacija uporabniku nudi opise kolesarskih in peš poti, točk na posamezni poti in znamenitosti. Ko uporabnik izbere eno od poti, ga aplikacija z glasovnimi navodili vodi po tej poti, kar je še posebej uporabno pri vožnji s kolesom. Na poti mu tudi zvočno opiše kakšno bližnjo naravno ali kulturno znamenitost.

To aplikacijo smo naredili za platformo Android s pomočjo IDE Visual Studio in vtičnika Xamarin, ki je namenjen za ustvarjanje mobilnih aplikacij za Android in iOS. Za upravljanje z zemljevidom smo uporabili Google Maps API. Aplikacija prikaže tudi trenutne GPS koordinate in lokacijo na zemljevidu, na katerem prikaže tudi točke na poti. Aplikacijo smo tudi testirali v praksi in smo z njenim delovanjem zadovoljni.

8. ZAHVALE

Zahvaljujemo se našemu mentorju Gregorju Hrastniku za pomoč in podporo pri raziskovalni nalogi, kot tudi sošolcu Žigi Meži, ki je za nas posnel glasove vodenja. Hvala dr. Nataši Meh Peer za lektoriranje naloge in Islamu Mušiču za začetno pomoč pri raziskovalni nalogi.

VIRI IN LITERATURA

- [1] Mobile app: https://en.wikipedia.org/wiki/Mobile_app, 10. 2. 2020
- [2] Aleš Rosina: Lokacijske storitve na mobilnih napravah: <http://eprints.fri.uni-lj.si/1632/1/Rosina1.pdf>, 10. 2. 2020
- [3] Mobile app development: https://en.wikipedia.org/wiki/Mobile_app_development, 10. 2. 2020
- [4] “Zombie” Apps On The Rise – 83% Of Apps Not On Top Lists, Up From 74% Last Year : <https://techcrunch.com/2015/01/30/zombie-apps-on-the-rise-83-of-apps-not-on-top-lists-up-from-74-last-year/>, 10. 2. 2020
- [5] Microsoft Store (digital): [https://en.wikipedia.org/wiki/Microsoft_Store_\(digital\)](https://en.wikipedia.org/wiki/Microsoft_Store_(digital)), 10. 2. 2020
- [6] Global Positioning System: https://en.wikipedia.org/wiki/Global_Positioning_System#External_links , 10. 2. 2020
- [7] Google Zemljevidi: https://sl.wikipedia.org/wiki/Google_Zemljevidi, 10. 2. 2020
- [8] Text-to-Speech Technology: What It Is and How It Works: <https://www.understood.org/en/school-learning/assistive-technology/assistive-technologies-basics/text-to-speech-technology-what-it-is-and-how-it-works>, 10. 2. 2020
- [9] Apple Maps: https://en.wikipedia.org/wiki/Apple_Maps, 10. 2. 2020
- [10] Visual Studio: <https://visualstudio.microsoft.com/vs/>, 10. 2. 2020
- [11] Xamarin: <https://dotnet.microsoft.com/apps/xamarin>, 10. 2. 2020
- [12] Xamarin Forms dokumentacija: <https://docs.microsoft.com/sl-si/xamarin/get-started/>, 10. 2. 2020
- [13] Geo awesomeness: <https://geoawesomeness.com/new-broadcom-gps-chip-will-make-maps-lbs-more-accurate/>, 10. 2. 2020
- [14] Apple Support: <https://support.apple.com/sl-si/HT202570>, 10. 2. 2020
- [15] MS Power User: <https://mspoweruser.com/new-xamarin-live-player-makes-it-easier-to-develop-ios-and-android-apps-on-windows/>, 10. 2. 2020