



Where2P4Free

Bachelor of Science (Honours) Software Development

Name: Patrick Browne

ID: C00190601

Year: 4th Year

Supervisor: Paul Barry

Due date: 12-04-2019

Technical Manual

Table of Contents

1. Introduction	2
2. Installation Requirements	3
2.1 Prerequisites	3
2.1 Installation	3
3. Ionic Source Code	4
3.1 Directions-tab.page.html	4
3.2 Directions-tab.page.scss	4
3.3 Directions-tab.page.ts	4
3.4 Home.module.ts	12
3.5 Home.page.html.....	13
3.6 Restrooms-tab.page.html.....	13
3.7 Restrooms-tab.page.ts.....	14
3.8 Rest-service.ts	14
3.9 App-routing.module.ts	16
3.10 App.component.html	17
4.0 Flask/API Source Code	17

1. Introduction

The purpose of this document is to provide instructions on how to install the mobile app on a device. All source code from the project is also included.

2. Installation Requirements

2.1 Prerequisites

In order to install the APK file you must ensure that third-party apps can be installed. To do this, go to **Settings > Security >** and check **Unknown Sources**. This will allow apps from sources other than the Google Play Store to be installed. Connect your Android device to your computer to transfer the file across.

2.2 Installation

Download the APK file from the website at the following link:. Copy the APK file from your computer to the device. Locate the file on the device, tap it, then click install.

3. Ionic Source Code

3.1 Directions-tab.page.html

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>Where2P4Free</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div id="map_canvas">

  </div>
</ion-content>
```

3.2 Directions-tab.page.scss

```
#map_canvas {
  height: 100%;
}
```

3.3 Directions-tab.page.ts

```
import { Component, OnInit } from '@angular/core';
import { GoogleMaps, GoogleMap, GoogleMapsEvent, Marker, MyLocation,
Environment, HtmlInfoWindow, GoogleMapsAnimation, LatLng, MarkerOptions,
GoogleMapOptions, LocationService } from '@ionic-native/google-maps/ngx';
import { Platform } from '@ionic/angular';
import { RestService } from '../..service/rest.service';

@Component({
  selector: 'app-directions-tab',
  templateUrl: './directions-tab.page.html',
  styleUrls: ['./directions-tab.page.scss'],
})
export class DirectionsTabPage {

  map: GoogleMap;
```

```

locations: any;
error: any;
location: any;
toiletID: any;
updatedLocation: any;
pointOfInterest: any;
variable = false;
latLng: LatLng;

constructor(private platform: Platform, public restService: RestService)
{
    if (this.variable = false) {
        this.variable = true;
    }
    else {
        this.getLocations();
    }
}

goToMyLocation() {
    this.map.getMyLocation().then((location: MyLocation) => {

        this.map.moveCamera({
            target: location.latLng,
            zoom: 18,
        });

        let marker: Marker = this.map.addMarkerSync({
            icon: 'blue',
            position: location.latLng,
        });

        let htmlInfoWindow = new HtmlInfoWindow();

        let frame: HTMLElement = document.createElement('div');
        frame.innerHTML = [
            `
                <p>Current Location</p>
                <form>
                    <ion-button type="submit" class="refreshMap">Refresh</ion-
button>
                </form>
            `
        ].join("");
    });
}

```

```

frame.getElementsByClassName("refreshMap")[0].addEventListener("click", ()
=> {
    console.log('Map refreshed!');
});

htmlInfoWindow.setContent(frame, {
    width: "125px",
    height: "125px"
});

htmlInfoWindow.open(marker);

this.map.on(GoogleMapsEvent.MAP_READY).subscribe(
    (data) => {
        console.log("Click MAP",data);
    }
);
})
}

getLocations() {
    this.restService.getLocations().then(data => {
        this.locations = data;
        for (let i = 0; i < this.locations.length; i++) {
            let marker: Marker = this.map.addMarkerSync({
                position: {
                    lat: this.locations[i]['latitude'],
                    lng: this.locations[i]['longitude'],
                },
            });
        }
    });

    let htmlInfoWindow = new HtmlInfoWindow();

    let frame: HTMLElement = document.createElement('div');
    frame.innerHTML = [
        `

` + this.locations[i]['location'] + `</p>` ,
        `
            <ion-button class="updateMarker">Update</ion-button>
            <ion-button class="removeMarker">Remove</ion-button>
        `
    ].join("");
}


```

```

frame.getElementsByClassName("updateMarker")[0].addEventListener("click",
() => {
    let htmlInfoWindow = new HtmlInfoWindow();

    let frame: HTMLElement = document.createElement('div');
    frame.innerHTML = [
        <p>Enter details below:</p>
        <label>Location:</label>
        <p><input type="text" id="location" name="location"
required></p>
        <label>Description:</label>
        <p><input type="text" id="description" name="description"
required></p>
        <label>Type:</label>
        <p><select id="typeList"></p>
            <option></option>
            <option>Male</option>
            <option>Female</option>
            <option>Gender Neutral</option>
            <option>Disabled</option>
        </select>
        <p><ion-button class="updateRestroom">Update</ion-button>
        <ion-button class="closeInfoWindow">Close</ion-button></p>
    ].join(",");

frame.getElementsByClassName("updateRestroom")[0].addEventListener("click"
, () => {
    var location =
(<HTMLInputElement>document.getElementById("location")).value;
    var description =
(<HTMLInputElement>document.getElementById("description")).value;
    var type =
(<HTMLSelectElement>document.getElementById("type")).value;

    var updateRestroomLocation = {};

    updateRestroomLocation['toiletID'] =
this.locations[i]['toiletID'];
    updateRestroomLocation['location'] = location;
    updateRestroomLocation['description'] = description;
    updateRestroomLocation['type'] = type;

```



```

        updateRestroomLocation['latitude'] =
this.locations[i]['latitude'];
        updateRestroomLocation['longitude'] =
this.locations[i]['longitude'];

this.restService.updateLocation(updateRestroomLocation).then((result) => {
    console.log(result);
    }, (err) => {
    console.log(err);
    });
});

frame.getElementsByClassName("closeInfoWindow")[0].addEventListener("click", () => {
    htmlInfoWindow.close();
});

htmlInfoWindow.setContent(frame, {
    width: "275px",
    height: "325px"
});

htmlInfoWindow.open(marker);
});

frame.getElementsByClassName("removeMarker")[0].addEventListener("click",
() => {
    var removeMarkerLocation = {};
    removeMarkerLocation['toiletID'] =
this.locations[i]['toiletID'];

this.restService.removeLocation(removeMarkerLocation).then((result) => {
    console.log(result);
    }, (err) => {
    console.log(err);
    });

    marker.remove();
});

htmlInfoWindow.setContent(frame, {

```

```

        width: "275px",
        height: "125px"
    });

    marker.on(GoogleMapsEvent.MARKER_CLICK).subscribe(() => {
        htmlInfoWindow.open(marker);
    });
}
});
}

addMarkerLocation() {
    this.map.on(GoogleMapsEvent.MAP_LONG_CLICK).subscribe(
        (data) => {
            this.location = data;
            for (let i = 0; i < this.location.length; i++) {
                let marker: Marker = this.map.addMarkerSync({
                    position: {
                        lat: this.location[i]['lat'],
                        lng: this.location[i]['lng'],
                    },
                });

                let htmlInfoWindow = new HtmlInfoWindow();

                let frame: HTMLElement = document.createElement('div');
                frame.innerHTML = [
                    `
                    <p>Marker added. Do you wish to add this restroom</p>
                    <ion-button class="addMarker">Add</ion-button>
                    <ion-button class="removeMarker">Remove</ion-button>
                `
                ].join("");

                frame.getElementsByClassName("addMarker")[0].addEventListener("click", ()
=> {
                    let htmlInfoWindow = new HtmlInfoWindow();

                    let frame: HTMLElement = document.createElement('div');
                    frame.innerHTML = [
                        `
                        <p>Enter information below:</p>
                        <label>Location:</label>
                        <p><input type="text" id="location" name="location"></p>
                    `
                ]

```

```

        <label>Description:</label>
        <p><input type="text" id="description"
name="description"></p>
        <label>Type:</label>
        <p><select id="typeList"></p>
            <option></option>
            <option>Male</option>
            <option>Female</option>
            <option>Gender Neutral</option>
            <option>Disabled</option>
        </select>
        <p><ion-button class="addRestroom">Submit</ion-button>
        <ion-button class="closeInfoWindow">Close</ion-button></p>
    `
    ].join("");

frame.getElementsByClassName("addRestroom")[0].addEventListener("click",
() => {
    var location =
(<HTMLInputElement>document.getElementById("location")).value;
    var description =
(<HTMLInputElement>document.getElementById("description")).value;
    var type =
(<HTMLSelectElement>document.getElementById("typeList")).value;

    var addRestroomLocation = {};

    addRestroomLocation['location'] = location;
    addRestroomLocation['description'] = description;
    addRestroomLocation['type'] = type;
    addRestroomLocation['lat'] = this.location[i]['lat'];
    addRestroomLocation['lng'] = this.location[i]['lng'];

this.restService.addLocation(addRestroomLocation).then((result) => {
    console.log(result);
    }, (err) => {
    console.log(err);
    });
});

frame.getElementsByClassName("closeInfoWindow")[0].addEventListener("click",
(), => {

```

```

        console.log('InfoWindow closed!');

        htmlInfoWindow.close();
    });

    htmlInfoWindow.setContent(frame, {
        width: "275px",
        height: "325px"
    });

    htmlInfoWindow.open(marker);
});

frame.getElementsByClassName("removeMarker")[0].addEventListener("click",
() => {
    console.log('Marker removed!');

    marker.remove();
});

htmlInfoWindow.setContent(frame, {
    width: "275px",
    height: "125px"
});

marker.on(GoogleMapsEvent.MARKER_CLICK).subscribe(() => {
    htmlInfoWindow.open(marker);
});
}
}
);
}

async loadMap() {
    Environment.setEnv({
        'API_KEY_FOR_BROWSER_RELEASE': '',
        // 'API_KEY_FOR_BROWSER_DEBUG':
        'AIzaSyAtcaMeDtBfN4lc6aYcL2gZxCoLejUj_tc'
    });
    this.map = GoogleMaps.create('map_canvas');
    this.goToMyLocation();
    this.addMarkerLocation();
    // this.addPointOfInterest();
}
}

```

```

    async ngOnInit() {
      await this.platform.ready();
      await this.loadMap();
    }
  }
}

```

3.4 Home.module.ts

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Routes, RouterModule } from '@angular/router';

import { IonicModule } from '@ionic/angular';

import { HomePage } from './home.page';

const routes: Routes = [
  {
    path: '',
    component: HomePage,
    children: [
      {
        path: 'directionstab',
        loadChildren: '../directions-tab/directions-
tab.module#DirectionsTabPageModule'
      },
      {
        path: 'restroomstab',
        loadChildren: '../restrooms-tab/restrooms-
tab.module#RestroomsTabPageModule'
      },
      {
        path: 'directionstab/addrestroom',
        loadChildren: '../add-restroom/add-
restroom.module#AddRestroomPageModule'
      }
    ]
  },
  {
    path: '',
    redirectTo: 'directions/directionstab',

```

```

    pathMatch: 'full'
  }
];

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RouterModule.forChild(routes)
  ],
  declarations: [HomePage]
})
export class HomePageModule {}

```

3.5 Home.page.html

```

<ion-tabs>
  <ion-tab-bar slot="bottom">

    <ion-tab-button tab="directionstab">
      <ion-icon name="navigate"></ion-icon>
      <ion-label>Directions</ion-label>
    </ion-tab-button>

    <ion-tab-button tab="restroomstab">
      <ion-icon name="body"></ion-icon>
      <ion-label>Restrooms</ion-label>
    </ion-tab-button>

  </ion-tab-bar>
</ion-tabs>

```

3.6 Restrooms-tab.page.html

```

<ion-header>
  <ion-toolbar color="primary">
    <ion-title>Where2P4Free</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

```

```

<ion-item lines="none" *ngFor="let toilet of toilets">
  <ion-card>
    <ion-card-content>
      <h2>Location: {{toilet.location}}</h2>
      <p>Description: {{toilet.description}}</p>
      <p>Type: {{toilet.type}}</p>
    </ion-card-content>
  </ion-card>
</ion-item>
</ion-content>

```

3.7 Restrooms-tab.page.ts

```

import { Component } from '@angular/core';
import { RestService } from '../..../service/rest.service';

@Component({
  selector: 'app-restrooms-tab',
  templateUrl: './restrooms-tab.page.html',
  styleUrls: ['./restrooms-tab.page.scss'],
})
export class RestroomsTabPage {

  toilets: any;

  constructor(public restService: RestService) {
    this.getLocations();
  }

  getLocations() {
    this.restService.getLocations().then(data => {
      this.toilets = data;
    });
  }
}

```

3.8 Rest-service.ts

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

```

```

import { HttpHeaders } from '@angular/common/http';

const httpOptions = {
  headers: new HttpHeaders({
    'Content-Type': 'application/json',
    'Authorization': 'my-auth-token'
  })
};

@Injectable({
  providedIn: 'root'
})
export class RestService {

  apiUrl = 'http://where2p4free.pythonanywhere.com/api';

  constructor(public http: HttpClient) {
    console.log('Hello RestServiceProvider Provider');
  }

  getLocations() {
    return new Promise(resolve => {
      this.http.get(this.apiUrl + "/findLocation").subscribe(data => {
        resolve(data);
      }, err => {
        console.log(err);
      });
    });
  }

  addLocation(data) {
    console.log(data);
    return new Promise((resolve, reject) => {
      this.http.post(this.apiUrl + "/addLocation", JSON.stringify(data),
httpOptions).subscribe(res => {
        resolve(res);
      }, (err) => {
        reject(err);
      });
    });
  }

  updateLocation(data) {
    console.log(data);
    return new Promise((resolve, reject) => {

```



```

        this.http.post(this.apiUrl + "/updateLocation",
JSON.stringify(data), httpOptions).subscribe(res => {
    resolve(res);
}, (err) => {
    reject(err);
});
});
}

removeLocation(data) {
    console.log(data);
    return new Promise((resolve, reject) => {
        this.http.post(this.apiUrl + "/removeLocation",
JSON.stringify(data), httpOptions).subscribe(res => {
            resolve(res);
        }, (err) => {
            reject(err);
        });
    });
}
}
}

```

3.9 App-routing.module.ts

```

import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [
    { path: '', redirectTo: 'directions/directionstab', pathMatch: 'full' },
    { path: 'directions', loadChildren:
'./pages/home/home.module#HomePageModule' },
];

@NgModule({
    imports: [
        RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules
    })
    ],
    exports: [RouterModule]
})
export class AppRoutingModule { }

```

3.10 App.component.html

```
<ion-app>
  <ion-router-outlet></ion-router-outlet>
</ion-app>
```

4. Flask/API Source Code

```
from flask import Flask, jsonify, request
from flask_cors import CORS
import mysql.connector

app = Flask(__name__)
CORS(app)

addThreshold = 2
updateThreshold = 2
removeThreshold = 2

@app.route('/')
def hello_world():
    return 'Hello from Flask!'

@app.route('/api/findLocation', methods=['GET'])
def retrieve_location():
    cnx = mysql.connector.connect(user="where2p4free",
password="restroom", host="where2p4free.mysql.pythonanywhere-
services.com", database="where2p4free$where2p4free")
    cursor = cnx.cursor(buffered=True)

    cursor.execute("SELECT toiletID, location, description, type,
latitude, longitude FROM toilet")
    cnx.commit()

    data = cursor.fetchall()

    d = []

    for location in data:
```

```

        d.append({'toiletID': location[0],
                'location': location[1],
                'description': location[2],
                'type': location[3],
                'latitude': location[4],
                'longitude': location[5]})
    cursor.close()
    cnx.close()

    return jsonify(d)

@app.route('/api/addLocation', methods=['POST'])
def insert_location():
    cnx = mysql.connector.connect(user="where2p4free",
    password="restroom", host="where2p4free.mysql.pythonanywhere-
    services.com", database="where2p4free$where2p4free")
    cursor = cnx.cursor()

    data = request.get_json()
    # print(data)

    sql = "INSERT INTO toilet (location, description, type, latitude,
    longitude) VALUES (%s, %s, %s, %s, %s)"
    cursor.execute(sql, (data['location'], data['description'],
    data['type'], data['lat'], data['lng']))
    cnx.commit()

    cursor.close()
    cnx.close()

    return 'Location added!'

@app.route('/api/updateLocation', methods=['POST'])
def update_location():
    cnx = mysql.connector.connect(user="where2p4free",
    password="restroom", host="where2p4free.mysql.pythonanywhere-
    services.com", database="where2p4free$where2p4free")
    cursor = cnx.cursor(buffered=True)

    data = request.get_json()

    sql = "SELECT toiletID, updateThreshold FROM toilet WHERE toiletID =
    %s"
    cursor.execute(sql % data['toiletID'])
    cnx.commit()

    values = cursor.fetchall()

```

```

threshold = values[0][1]

if threshold < updateThreshold:
    sql = "UPDATE toilet SET updateThreshold = updateThreshold + 1
WHERE toiletID = %s"
    cursor.execute(sql % data['toiletID'])
    cnx.commit()
    s = 'Location cannot yet be update!'
else:
    sql = "UPDATE toilet SET location = %s, description = %s, type =
%s, latitude = %s, longitude = %s, updateThreshold = 0 WHERE toiletID =
%s"
    cursor.execute(sql, (data['location'], data['description'],
data['type'], data['latitude'], data['longitude'], data['toiletID']))
    cnx.commit()
    s = 'Location updated!'

cursor.close()
cnx.close()

return s

@app.route('/api/removeLocation', methods=['POST'])
def remove_location():
    cnx = mysql.connector.connect(user="where2p4free",
password="restroom", host="where2p4free.mysql.pythonanywhere-
services.com", database="where2p4free$where2p4free")
    cursor = cnx.cursor(buffered=True)

    data = request.get_json()

    sql = "SELECT toiletID, deleteThreshold FROM toilet WHERE toiletID =
%s"
    cursor.execute(sql % data['toiletID'])
    cnx.commit()

    values = cursor.fetchall()
    threshold = values[0][1]

    if threshold < removeThreshold:
        sql = "UPDATE toilet SET deleteThreshold = deleteThreshold + 1
WHERE toiletID = %s"
        cursor.execute(sql % data['toiletID'])
        cnx.commit()
        s = 'Location cannot yet be removed!'
    else:
        sql = "DELETE FROM toilet WHERE toiletID = %s"

```

```
    cursor.execute(sql % data['toiletID'])
    cnx.commit()
    s = 'Location removed!'

cursor.close()
cnx.close()

return s
```