FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT

LUCRAREA DE LABORATOR#4

Dezvoltarea unei aplicatii mobile

Autor:

Lașcu Mihai

Lector asistent:

Cojanu Irina

Lector superior:

Radu Melnic

Lucrarea de laborator #4

1 Obiectivele lucrarii

Cunostintele de baza privina arhitectura unei aplicatii mobile Cunostintele de baza ale platformei SDK

2 Scopul lucrarii de laborator

Realizarea aplicatie *MAP*(googleMap) pe **IOS**

Geolocalizarea pe MAP

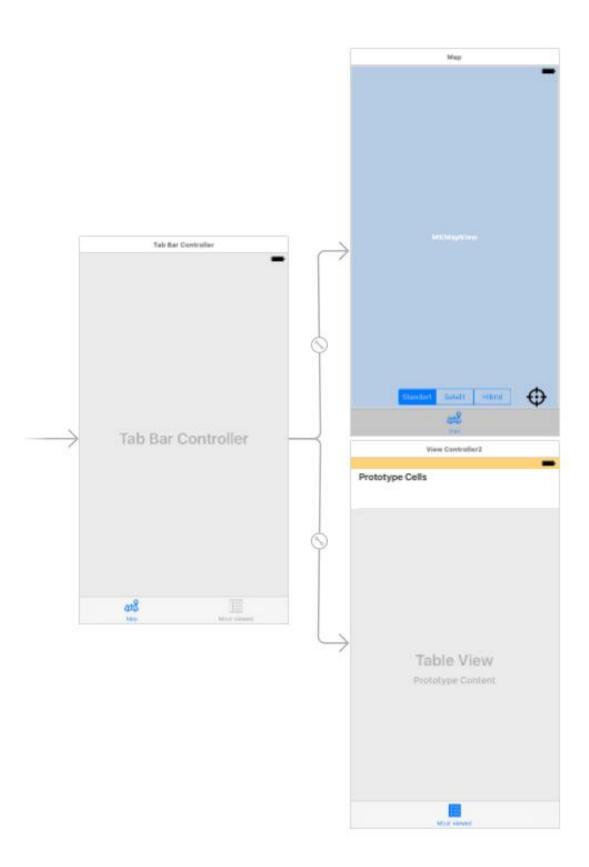
SegmentControlBar reprezentarea modurilor pe *MAP*(Hybrid, Satellite si Standart)

Tab Bar reprezentarea a 2 icoane(Map,List)

Table View lista locatiilor care directioneaza la penitele plaste pe mapa Adaugarea constraint-urilor pentru afisarea corecta in pozitie orizontala a aplicatiei

3 Mersul lucrarii de laborator

Drept IDE am folosit XCode3. Ca limbaj de programare a fost folosit Swift. Aplicatia data nu este Cross-Platform. Aplicatia are 2 View-uri. Primul View este redat instrumentul MapKit.Al doilea View este redata lista de locatii ale aplicatiei care pot fi accesate.



#View 1

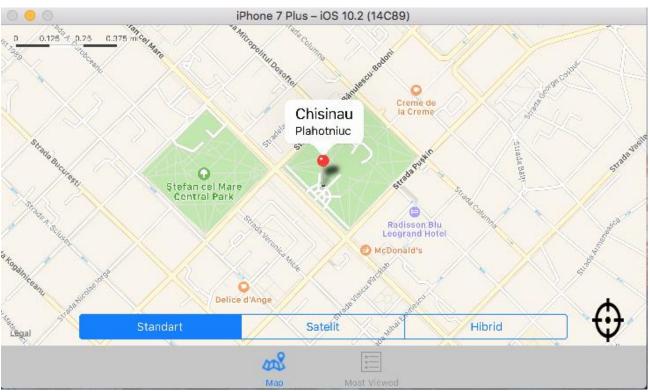
```
import UIKit
import MapKit
import CoreLocation
class ViewController: UIViewController, MKMapViewDelegate, CLLocationManagerDelegate {
    @IBOutlet weak var mapView: MKMapView!
    var locationManager = CLLocationManager()
    var indice = -1
    var region: MKCoordinateRegion?
    override func viewDidLoad() {
        super.viewDidLoad()
        locationManager.requestWhenInUseAuthorization()
        locationManager.delegate = self
        locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters
        locationManager.distanceFilter = 10
        locationManager.startUpdatingLocation()
        displayMap()
    func displayMap() {
        for j in 0..<name.count {
            let annotation = MKPointAnnotation()
            let latitude = name[j].x
            let longitude = name[j].y
            let location = CLLocationCoordinate2DMake(latitude, longitude)
            annotation.coordinate = location
            annotation.title = name[j].name
            annotation.subtitle = name[j].desc
            let span = MKCoordinateSpanMake(0.009, 0.009)
            let region = MKCoordinateRegion(center: location, span: span)
            self.mapView.addAnnotation(annotation)
            if indice == j || (j == 0 && indice == -1){
    self.mapView.setRegion(region, animated: true)
                self.mapView.selectAnnotation(annotation, animated: true)
        }
    }
    @IBAction func ACTION(_ sender: UISegmentedControl) {
        switch (sender.selectedSegmentIndex) {
        case 0:
            mapView.mapType = MKMapType.standard
        case 1:
            mapView.mapType = MKMapType.satellite
        default:
            mapView.mapType = MKMapType.hybrid
    }
    @IBAction func SetLocation(_ sender: UIButton) {
         mapView.setRegion(region!, animated: true)
    func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
        let location = locations[0]
        let span:MKCoordinateSpan = MKCoordinateSpanMake(0.01, 0.01)
        let myLocation = CLLocationCoordinate2DMake(location.coordinate.latitude, location.coordinate.longitude)
        region = MKCoordinateRegionMake(myLocation ,span)
       // mapView.setRegion(region, animated: true)
        mapView.showsUserLocation = true
        print ("locatie gasita")
}
```

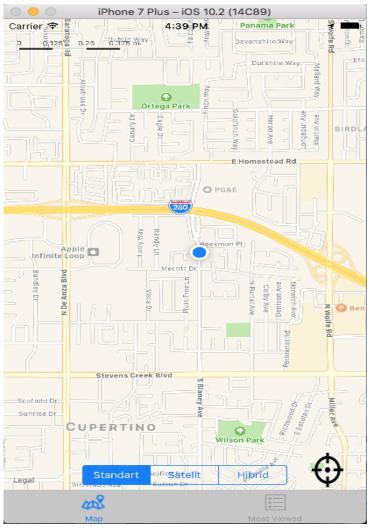
#View2

```
// ViewController2.swift
// D05
//
// Created by mihai lascu on 4/28/17.
// Copyright @ 2017 mihai_lascu. All rights reserved.
//
//
import UIKit
var name = [(name: "Ecole 42", desc: "Academy", x: 48.8965812, y: 2.31837599999944),
(name: "Chisinau", desc: "Plahotniuc", x: 47.025136, y: 28.832911),
(name: "Paris", desc: "Fance", x: 48.857953, y: 2.342706),
(name: "Londra", desc: "Leps", x: 51.511041, y: -0.121537)]
//let color:[UIColor] = [.red, .yellow, .green, .blue]
class ViewController2: UIViewController, UITableViewDelegate, UITableViewDataSource {
   @IBOutlet weak var tableView1: UITableView!
    override func viewDidLoad() {
        super.viewDidLoad()
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        let barViewControllers = self.tabBarController?.viewControllers
        let svc = barViewControllers![0] as! ViewController
        svc.indice = indexPath.row
        svc.displayMap()
        tabBarController?.selectedIndex = 0
    ////
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return name.count
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = self.tableView1.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
        cell.textLabel?.text = name[indexPath.row].name
        return cell
   }
}
```

4 App Screenshots











Concluzie

In lucrarea data s-a creat o aplicatie mobile pe **IOS**. Am folosit un **Table View Controller** ce ne usureaza lucrul cu 2 View-uri.Primu *View* afiseaza *MapKit-ul* segmentul de moduri ale mapei si butonul *geolocation*. Al doilea View afiseaza lista locatiilor care au fost setate dupa latitudine si longitudine, iar cind apasm pe ele ne directioneaza direct la coordonatele indicate.Am folosit Constraint-urile pentru a avea o pozitie corecta cind vom intoarce dispozitivul in pozitie orizontala. Ca IDE s-a folosit **XCode 3.** In urma lucrarii am acumulat multa experienta pe mobile, si am invatat limbajul **Swift.**