Carbook документација

# УИ контрола по свој дизајн

На почетокот при внесување на нова кола, копчето што е дел од Navigation Bar е со слика во личен дизајн.

Исто така секое копче од Bar-от кај мапите е од пакет симнат преку интернет со бесплатна лиценца за дистрибуција во искористување на сликите со личен дизајн. Откако ќе се одбере локација и потоа селекција на првото копче во Bar-от (што изгледа како Favorites) се добиваат Table Cells направени со модел и дизајн изработен различно од понудениот во XCode преку финкцијата

- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

во CBStepsTableViewController.

Следна контрола се MKPinAnnotationView за најблиските локации чиј изглед е потпомогнат од страна на вградената енумерација *UIButtonTypeInfoLight* така што при клик се добива информација за патеката од моменталната локација која се наоѓа во CBMapSampleViewController.m во функцијата

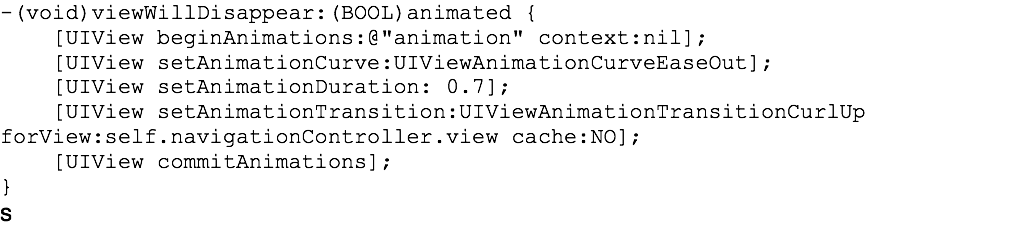
- (MKAnnotationView \*)mapView:(MKMapView \*)mapView viewForAnnotation:(id <MKAnnotation>)annotation

Исто така употребуваме слоеви над мапата за цртање на патеката од моменталната локација до крајната дестинација со употреба на MKPolylineRenderer класата во функцијата

- (MKOverlayRenderer \*)mapView:(MKMapView \*)mapView rendererForOverlay:(id < MKOverlay >)overlay

од CBMapSampleViewController.m.

# Анимација на УИ по сопствен дизајн

Ние користиме Navigation Controller за да влегуваме/излегуваме од сите контролери. Притоа при секој Segue користиме анимацијата UIViewAnimationTransitionCurlUp со брзина од 0.7 секунди и UIViewAnimationCurveEaseOut за да се намалува при крајот. Следниот код моментално е поставен во CBMapSampleViewController.m.

# Локациски сервиси и мапи

Во нашата апликација ги употребуваме Apple Maps со вградениот Location Manager кој успева да ја одредува локацијата според тоа што му е на располагање. Доколку корисникот има вклучено GPS, ќе употребува GPS на секои 100 метри. Ако во близина постојат WiFi hotspots и корисникот има вклучено WiFi, тогаш со цел да се заштеди на батерија ќе употребува WiFi. Притоа се употребуваат MKMapViewDelegate и CLLocationManagerDelegate протоколите за функционалноста на Епл мапите и локациските сервиси во CBMapSampleViewController.m кои можат да се забележат во XCode во две секции: CLLocationManagerDelegate/MKMapViewDelegate methods

# Веб сервис

Се користат open-source веб сервисите на FuelEconomy.gov кои се воедно официјалните информации кои ги дава Американското Министерство за Економија. Податоците кои се парсираат се во XML формат и се во галони по милја потрошувачка на гориво, а за потребите на нашата апликација истите ги конвентираме во литри на 100 километри потрошувачка. Не се сите модели на автомобили опфатени, а исто така најстарите модели за кои има податоци се од 1970. Во самата апликација за да се одбере даден модел и да се добие неговата соодветна потрошувачка е потребно да се наведе точната негова година на производство бидејќи доколку се користи година во која не е направен самиот модел, сервисите враќаат празен XML фајл. Пример на автомобили кои се поддржани се многу, ќе наведам само 6 тука а доколку се провери соодветната година на производство на автомобилот ќе се добијат соодветни податоци. Тестирани модели на автомобили: Mitsubishi Lancer 2008, Mercedes SLK 2005, Audi A4 2012, Audi A6 2012, Aston Martin V8 Vantage 2007, Alfa Romeo 164 1993. Потрошувачката на овие модели која нашата апликација ја дава соодветствува со потрошувачката на повеќето интернет страници за автомобилизам како и интернет форуми.Овој дел од домашната се наоѓа во CBAppDelegate во методот pustiBaranje, но се повукува и CBrssPaser класата која претставува xml парсер.

# Два сензори по желба

При селекција на копчето за камера имаме употреба на сензорот на камерата кој се активира преку класата CBCameraSharer.m во нејзиниот конструктор повикан од функцијата useCamera во CBMapSampleViewController.m. Вториот сензор е магнетометарот преку употреба на класата *CLLocationManager* (која исто така ја користиме за локацијата) и ротација на мапата со помош на две прсти се добива информација во горниот десен агол за насоката на магнетните полови. Функционалноста е овозможена во CBMapSampleViewController.m во функциите

-(void)locationManager:(CLLocationManager \*)manager didUpdateHeading:(CLHeading \*)newHeading

-(void)locationManager:(CLLocationManager \*)manager didUpdateToLocation:(CLLocation \*)newLocation fromLocation:(CLLocation \*)oldLocation

# Интеграција со социјална мрежа

Изборот на социјална мрежа решивме да биде Facebook. Направена е апликација на Facebook преку која може да се користи Facebook SDK-то на iOS 7. Од нашата апликација при избор на иконата којашто ја имаме за оваа социјална мрежа, се прави Share на соодветната апликација како линк, а на корисникот му се овозможува да каже нешто повеќе околу искуството со оваа апликација, од типот колку поминати километри има или колкава му е потрошувачката на корисникот... Доколку апликацијата се постави на App Store во тој случај линкот кон апликацијата на Facebook би водел до App Store за симнување на апликацијата. Овој дел од кодот се наоѓа во CBMapSampleViewController поточно во методата shareImage.

# Внатрешен Storage

Направен е внатрешен storage кои го овозможува следново сценарио. Доколку корисникот одбере марка модел и година, и истиот добил валидни податоци за потрошувачката на гориво на неговиот автомобил, се зачувува фајл со име carbook.data во којшто се складира неговата потрошувачка како текст. Доколку корисникот ја исклучи апликацијата (или телефонот) и потоа повторно ја уклучи истата апликација и ваквиот фајл постои, корисникот ќе биде директно пренесен на погледот на којшто е мапата без да биде потребно повторно да внесува податоци за неговото возило. Секако од погледот на мапата постои варијанта истиот корисник да си го смени возилото. Фајлот се зачувува во истиот директориум како и апликацијата. Овој дел од кодот се наоѓа во CBViewController класата од која се повикуваат readFile за читање од локален storage како и методот prepareForSegue во којшто се извршува зачувување на потрошувачката пред да се преиде во следниот контролер.

# Multithreading

Во нашата апликација цело време добиваме информации преку мрежа. Како резултат на конекцијата може да настане замрзнување на UI поради синхрони операции. Следните функции употребуваат асинхрони повикувања:

* **CBMapSampleViewController.m**
  + (IBAction)shareImage:(id)sender во прикажување Share Dialog, Feed Dialog кој ќе работи асинхроно при прикачување на сликата
  + (void) performSearch кога бараме најблиски локации на мапа
  + (void)getDirections кога треба да ја најдеме патеката до одредена локација

# Нотификации

Направени се локални нотификации кои се активираат 10 минути откако корисникот ќе ја стави апликацијата во позадина, а декларирањето на ваквите нотификации се наоѓа во методот од AppDelegate кој се однесува за влегувањето на апликацијата во Background односно applicationDidEnterBackground. Избрани се 10 минути чисто заради полесен приказ за нотификациите, а времетраењето после кое ќе се прикаже нотификацијата е лесно нагодливо.

# Лиценца за мултимедија

Во нашата апликација Carbook користевме слики чиј дизајнер ги има пуштено во оптек со бесплатна лиценца. При употреба на Apple Maps нема потреба да додаваме атрибуција за нејзина употреба како резултат на вградената опција.