

```

//
// AppDelegate.m
// ShopAssists
//
// Created by NITS_Mac3 on 27/04/15.
// Copyright (c) 2015 NITS. All rights reserved.
//

#import "AppDelegate.h"
#import "MFSideMenu.h"
#import "MFSideMenu.h"
#import "ViewController.h"
#import "loadScreen.h"
#import "MainViewController.h"
#import "feed.h"
#import <EstimoteSDK/EstimoteSDK.h>
#import "AFNetworking.h"
#import "ConstantUrls.h"
#import "TSMMessage.h"
#import "TSMMessageView.h"
#import <TSMessages/TSMMessageView.h>
#import <CoreLocation/CoreLocation.h>
#define BEACON_1_UUID @"B9407F30-F5F8-466E-AFF9-25556B57FE6D"
#define BEACON_1_MAJOR 11889
#define BEACON_1_MINOR 17681

#define BEACON_2_UUID @"B9407F30-F5F8-466E-AFF9-25556B57FE6D"
#define BEACON_2_MAJOR 8318
#define BEACON_2_MINOR 47826

#define BEACON_3_UUID @"B9407F30-F5F8-466E-AFF9-25556B57FE6D"
#define BEACON_3_MAJOR 54093
#define BEACON_3_MINOR 23350

@interface AppDelegate
()<ESTBeaconConnectionDelegate, ESTBeaconManagerDelegate, CLLocationManagerDelegate >{
    CLLocation *nearestBeacon;
    CLLocationManager *locationManager;
    CLLocationRegion *region;
}
@property (nonatomic) ESTBeaconManager *beaconManager;
@property (nonatomic) CLLocationRegion *beaconRegion;
@property (nonatomic) CLLocationRegion *beaconRegion1;
@property (nonatomic) CLLocationRegion *beaconRegion2;
@property (nonatomic) CLLocationRegion *beaconRegion3;
@end

@implementation AppDelegate

```

```

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    [ESTCloudManager setupAppID:@"shopassist"
andAppToken:@"cb30b491cc695f0aca317535bd3ad7c6"];

    if ([[UIDevice currentDevice] systemVersion] floatValue] >= 8.0)
    {
        [[UIApplication sharedApplication]
registerUserNotificationSettings:[UIUserNotificationSettings
settingsForTypes:(UIUserNotificationTypeSound |
UIUserNotificationTypeAlert | UIUserNotificationTypeBadge)
categories:nil]];
        [[UIApplication sharedApplication]
registerForRemoteNotifications];
    }
    else
    {
        [[UIApplication sharedApplication]
registerForRemoteNotificationTypes:
(UIUserNotificationTypeBadge | UIUserNotificationTypeSound |
UIUserNotificationTypeAlert)];
    }
    NSUserDefaults *userData=[NSUserDefaults standardUserDefaults];
    NSDictionary *userData_=[userData objectForKey:@"data"];

    if (userData_ ==nil)
    {

        self.window.rootViewController = [[UIStoryboard
storyboardWithName:@"Main" bundle:[NSBundle mainBundle]]
instantiateInitialViewController];

    }
    else
    {
        MainViewController *rootController = [[UIStoryboard
storyboardWithName:@"Main" bundle:[NSBundle mainBundle]]
instantiateViewControllerWithIdentifier:@"MainViewController"];
        UINavigationController* navigation = [[UINavigationController
alloc] initWithRootViewController:rootController];
        self.window.rootViewController = navigation;

        UIStoryboard *storyboard = [UIStoryboard
storyboardWithName:@"Main" bundle:nil];
        MFSideMenuContainerViewController *container =
(MFSideMenuContainerViewController *)[storyboard
instantiateViewControllerWithIdentifier:@"MFSideMenuContainerViewContr

```

```

oller"];

        UINavigationController *navigationController = [storyboard
instantiateViewControllerWithIdentifier:@"navigationController"];

        UIViewController *leftSideMenuViewController = [storyboard
instantiateViewControllerWithIdentifier:@"leftSideMenuViewController"]
;
        [container
setLeftMenuViewController:leftSideMenuViewController];
        [container setCenterViewController:navigationController];
        self.window.rootViewController = container;

    }

    UILocalNotification *localNotif = [launchOptions
objectForKey:UIApplicationLaunchOptionsLocalNotificationKey];
    NSDictionary *dictionary = [launchOptions
objectForKey:UIApplicationLaunchOptionsRemoteNotificationKey];

    if (dictionary) {
        NSLog(@"Recieved Notification %@", localNotif);
        NSDictionary *aps = [dictionary objectForKey:@"aps"];
        NSUserDefaults *prefs = [NSUserDefaults standardUserDefaults];
        [prefs setObject:[aps objectForKey:@"tableID"]
forKey:@"localNotif"];
        [prefs synchronize];
    }

    self.beaconRegion1 = [[CLBeaconRegion alloc]
initWithProximityUUID:[NSUUID alloc]
initWithUUIDString:BEACON_1_UUID]

major:BEACON_1_MAJOR minor:BEACON_1_MINOR
identifier:@"beaconRegion1"];
    self.beaconRegion2 = [[CLBeaconRegion alloc]
initWithProximityUUID:[NSUUID alloc]
initWithUUIDString:BEACON_2_UUID]

major:BEACON_2_MAJOR minor:BEACON_2_MINOR
identifier:@"beaconRegion2"];
    self.beaconRegion3 = [[CLBeaconRegion alloc]
initWithProximityUUID:[NSUUID alloc]
initWithUUIDString:BEACON_3_UUID]

major:BEACON_3_MAJOR minor:BEACON_3_MINOR
identifier:@"beaconRegion3"];
    self.beaconManager = [ESTBeaconManager new];
    self.beaconManager.delegate = self;

```

```

self.beaconManager.returnAllRangedBeaconsAtOnce = YES;
[self.beaconManager requestAlwaysAuthorization];

CLLocationManager *manager = [[CLLocationManager alloc] init];
manager.delegate = self;
NSUUID *proximityUUID = [[NSUUID alloc]
initWithUUIDString:@"B9407F30-F5F8-466E-AFF9-25556B57FE6D"];
region = [[CLBeaconRegion alloc]
initWithProximityUUID:proximityUUID identifier:@"Estimote Region"];

self.beaconRegion = [[CLBeaconRegion alloc] initWithProximityUUID:
[[NSUUID alloc] initWithUUIDString:BEACON_3_UUID]
identifier:@"beaconRegion"];
return YES;
}

- (void)LoggedIn{

// [self.window setRootViewController:nil];
[self.beaconManager
startRangingBeaconsInRegion:self.beaconRegion];
[locationManager startMonitoringForRegion:region];
NSMutableArray *notify;
NSUserDefaults *prefs=[NSUserDefaults standardUserDefaults];
[prefs setObject:notify forKey:@"notify"];
// [self.beaconManager
startRangingBeaconsInRegion:self.beaconRegion1];
// [self.beaconManager
startRangingBeaconsInRegion:self.beaconRegion2];
// [self.beaconManager
startRangingBeaconsInRegion:self.beaconRegion3];

UIStoryboard *storyboard = [UIStoryboard
storyboardWithName:@"Main" bundle:nil];
MFSideMenuContainerViewController *container =
(MFSideMenuContainerViewController *)[storyboard
instantiateViewControllerWithIdentifier:@"MFSideMenuContainerViewContr
oller"];

UINavigationController *navigationController = [storyboard
instantiateViewControllerWithIdentifier:@"navigationController"];

UIViewController *leftSideMenuViewController = [storyboard
instantiateViewControllerWithIdentifier:@"leftSideMenuViewController"]
;
[container setLeftMenuViewController:leftSideMenuViewController];
[container setCenterViewController:navigationController];

```

```

        self.window.rootViewController = container;
    }

- (void)Logout{
    [self.beaconManager stopRangingBeaconsInRegion:self.beaconRegion];
    [self.window setRootViewController:nil];
    NSUserDefaults *userData=[NSUserDefaults standardUserDefaults];
    [userData removeObjectForKey:@"data"];

    [userData synchronize];

    UIStoryboard *storyboard = [UIStoryboard
storyboardWithName:@"Main" bundle:nil];
    UINavigationController *navigationController = [storyboard
instantiateViewControllerWithIdentifier:@"LoginView"];
    self.window.rootViewController = navigationController;
}

- (void)applicationWillResignActive:(UIApplication *)application {
    // Sent when the application is about to move from active to
inactive state. This can occur for certain types of temporary
interruptions (such as an incoming phone call or SMS message) or when
the user quits the application and it begins the transition to the
background state.
    // Use this method to pause ongoing tasks, disable timers, and
throttle down OpenGL ES frame rates. Games should use this method to
pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application {
    // Use this method to release shared resources, save user data,
invalidate timers, and store enough application state information to
restore your application to its current state in case it is terminated
later.
    // If your application supports background execution, this method
is called instead of applicationWillTerminate: when the user quits.
}

- (void)applicationWillEnterForeground:(UIApplication *)application {
    // Called as part of the transition from the background to the
inactive state; here you can undo many of the changes made on entering
the background.
}

- (void)applicationDidBecomeActive:(UIApplication *)application {
    application.applicationIconBadgeNumber = 0;
}

```

```
- (void)applicationWillTerminate:(UIApplication *)application {
    // Called when the application is about to terminate. Save data if
    // appropriate. See also applicationDidEnterBackground:.
}
```

```
- (void)application:(UIApplication*)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData*)deviceToken
{
    NSLog(@"My token is: %@", deviceToken);
   NSUserDefaults *prefs = [NSUserDefaults standardUserDefaults];

    [prefs setObject:deviceToken forKey:@"deviceToken"];
    [prefs synchronize];
}
```

```
- (void)application:(UIApplication*)application
didFailToRegisterForRemoteNotificationsWithError:(NSError*)error
{
    NSLog(@"Failed to get token, error: %@", error);
}
```

```
- (NSDictionary *)parametersDictionaryFromQueryString:(NSString
*)queryString {

    NSMutableDictionary *md = [NSMutableDictionary dictionary];

    NSArray *queryComponents = [queryString
componentsSeparatedByString:@"&"];

    for(NSString *s in queryComponents) {
        NSArray *pair = [s componentsSeparatedByString:@"="];
        if([pair count] != 2) continue;

        NSString *key = pair[0];
        NSString *value = pair[1];

        md[key] = value;
    }

    return md;
}
```

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication annotation:
(id)annotation {
```

```

        if ([[url scheme] isEqualToString:@"myapp"] == NO) return NO;

        NSDictionary *d = [self parametersDictionaryFromQueryString:[url
query]];

        NSString *token = d[@"oauth_token"];
        NSString *verifier = d[@"oauth_verifier"];

        feed *vc = (feed *)[self window] rootViewController];
        [vc setOAuthToken:token oauthVerifier:verifier];

        return YES;
    }

- (void)application:(UIApplication*)application
didReceiveRemoteNotification:(NSDictionary*)userInfo
{
    NSLog(@"Received notification: %@", userInfo);

    NSString *alertView = [[userInfo valueForKey:@"aps"]
valueForKey:@"alert"];

    NSMutableArray *parts = [NSMutableArray arrayWithArray:[alertView
componentsSeparatedByString:@": "]];
    NSString *senderName = [parts objectAtIndex:0];

    [parts removeObjectAtIndex:0];

    NSString *message = [parts componentsJoinedByString:@": "];

    [TSMMessage showNotificationWithTitle:senderName
subtitle:message type:TSMMessageNotificationTypeMessage];
}

BOOL isBeaconWithUUIDMajorMinor(CLBeacon *beacon, NSString
*UUIDString, CLBeaconMajorValue major, CLBeaconMinorValue minor) {

    return [beacon.proximityUUID.UUIDString
isEqualToString:UUIDString] && beacon.major.unsignedShortValue ==
major && beacon.minor.unsignedShortValue == minor;
}

-(void)beaconManager:(id)manager didRangeBeacons:(NSArray *)beacons
inRegion:(CLBeaconRegion *)region {

```

```

    nearestBeacon = [beacons firstObject];

    if (nearestBeacon) {
//      NSString *beacon=[[NSString alloc]
initWithFormat:@"%@",nearestBeacon];
//      UIAlertView *alert = [[UIAlertView alloc]
initWithTitle:@"Beacon found"
//
//      message:
beacon
//
//      delegate:nil
//
cancelButtonTitle:@"OK"
//
otherButtonTitles:nil];
//      [alert show];
      NSLog(@"%@",nearestBeacon);
      [self sendNotification];

      AFHTTPRequestOperationManager *manager =
[AFHTTPRequestOperationManager manager];

      NSString *strMsg = [[NSString alloc]
initWithFormat:@"%@",nearestBeacon];

      NSDictionary *parameters=
@{@"email":@"pss.manas@gmail.com" ,@"subject": @"Your ShopsAssist
List", @"content":strMsg};

      [manager POST:[NSString stringWithFormat:@"%@"
%@",BaseURL,ShareEmail] parameters: parameters
success:^(AFHTTPRequestOperation *operation, id responseObject) {

          if ([[responseObject objectForKey:@"Success"]
isEqualToString:@"1"]) {

              }

          } failure:^(AFHTTPRequestOperation *operation, NSError *error)
{
      NSLog(@"Error: %@", error);
    }
  }];
}
}

```



```

- (void)beaconManager:(id)manager didChangeAuthorizationStatus:
(CLAuthorizationStatus)status {
    if (status == kCLAuthorizationStatusDenied || status ==
kCLAuthorizationStatusRestricted) {
        NSLog(@"Location Services authorization denied, can't range");

        AFHTTPRequestOperationManager *manager =
[AFHTTPRequestOperationManager manager];

        NSString *strMsg = [[NSString alloc] initWithFormat:@"%d
-----> Location Services authorization denied, can't
range",status];

        NSDictionary *parameters=
@{@"email":@"pss.manas@gmail.com" ,@"subject": @"Your ShopsAssist
List", @"content":strMsg};

        [manager POST:[NSString stringWithFormat:@"%@"
%@",BaseURL,ShareEmail] parameters: parameters
success:^(AFHTTPRequestOperation *operation, id responseObject) {

            if ([[responseObject objectForKey:@"Success"]
isEqualToString:@"1"]) {

                }

            } failure:^(AFHTTPRequestOperation *operation, NSError *error)
{
                NSLog(@"Error: %@", error);

            }]];
    }
}

- (void)beaconManager:(id)manager rangingBeaconsDidFailForRegion:
(CLBeaconRegion *)region withError:(NSError *)error {

    // NSLog(@"Ranging beacons failed for region '%@\n\nMake sure that
Bluetooth and Location Services are on, and that Location Services are
allowed for this app. Also note that iOS simulator doesn't support
Bluetooth.\n\nThe error was: %@", region.identifier, error);

    manager = [AFHTTPRequestOperationManager manager];

    NSString *strMsg = [[NSString alloc] initWithFormat:@"Ranging
beacons failed for region '%@\n\nMake sure that Bluetooth and
Location Services are on, and that Location Services are allowed for
this app. Also note that iOS simulator doesn't support Bluetooth.\n

```

```

\nThe error was: %@",region.identifier, error];

    NSDictionary *parameters=
    @{@"email":@"pss.manas@gmail.com" ,@"subject": @"Your ShopsAssist
    List", @"content":strMsg};

    [manager POST:[NSString stringWithFormat:@"%@
    %@","BaseURL,ShareEmail] parameters: parameters
    success:^(AFHTTPRequestOperation *operation, id responseObject) {

        if ([[responseObject objectForKey:@"Success"]
        isEqualToString:@"1"]) {

            }

        } failure:^(AFHTTPRequestOperation *operation, NSError *error) {
            NSLog(@"Error: %@", error);

        }];
    }

- (void)sendNotification{

   NSUserDefaults *prefs=[NSUserDefaults standardUserDefaults];

    NSMutableArray *notify= [prefs mutableArrayValueForKey:@"notify"];

    NSDate *date = [NSDate date];
    int diff = 40;

    for(int i=0;i<notify.count;i++){
        NSDictionary *dict=[notify objectAtIndex:i];
        if([[dict objectForKey:@"major"]isEqualToString:[NSString
        stringWithFormat:@"%@",nearestBeacon.major]] && [[dict
        objectForKey:@"minor"]isEqualToString: [NSString
        stringWithFormat:@"%@",nearestBeacon.minor]]){
            NSDate *lastDate=[dict objectForKey:@"currentTime"];
            NSTimeInterval distanceBetweenDates = [date
            timeIntervalSinceDate:lastDate];
            diff=distanceBetweenDates/60;

        }
    }

    if(diff >15 || diff==40){
        NSDictionary *notifyParameters= @{@"major": [NSString

```

```

stringWithFormat:@"%@",nearestBeacon.major],@"minor":[NSString
stringWithFormat:@"%@",nearestBeacon.minor],@"currentTime":date};
    [notify addObject:notifyParameters];
    [prefs setObject:notify forKey:@"notify"];

    AFHTTPRequestOperationManager *manager =
[AFHTTPRequestOperationManager manager];
    NSUserDefaults *userData=[NSUserDefaults
standardUserDefaults];
    NSDictionary *userData_=[userData objectForKey:@"data"];
    NSDictionary *parameters= @{@"userId":[userData_
objectForKey:@"user_id" ],@"major":nearestBeacon.major,@"minor":neares
tBeacon.minor};

    [manager POST:[NSString stringWithFormat:@"%@"
%@",BaseURL,beaconDetect] parameters: parameters
success:^(AFHTTPRequestOperation *operation, id responseObject) {

        if ([[responseObject objectForKey:@"Success"]
isEqualToString:@"1"]) {

            }

        } failure:^(AFHTTPRequestOperation *operation, NSError *error)
{
        NSLog(@"Error: %@", error);

    }];
}

@end

```