MAP -

@property (strong, nonatomic) CLLocationManager \*locationManager;

[self.locationManager requestWhenInUseAuthorization];

[self.locationManager requestAlwaysAuthorization];

// MARK: Co-ordinate looping

if(setFreeHour.count != 0){

for(int j=0;j<setFreeHour.count;j++){

NSDate \*date= [[NSDate alloc] init];

NSDateFormatter \*dateFormatter = [[NSDateFormatter alloc]init];

NSTimeZone \*gmttoday = [NSTimeZone timeZoneForSecondsFromGMT:19800];

[dateFormatter setTimeZone:gmttoday];

[dateFormatter setDateFormat:@"yyyy-MM-dd hh:mm:ss"];

NSArray \*fromD = [setFreeHour valueForKey:@"fromDate"];

NSString \*fromdate = fromD[j];

NSDateFormatter \*fromDateFormatter = [[NSDateFormatter alloc]init];

NSTimeZone \*gmtfrom = [NSTimeZone timeZoneForSecondsFromGMT:19800];

[fromDateFormatter setTimeZone:gmtfrom];

[fromDateFormatter setDateFormat:@"yyyy-MM-dd hh:mm:ss"];

NSDate \*fromDate = [fromDateFormatter dateFromString:fromdate];

NSArray \*toD = [setFreeHour valueForKey:@"toDate"];

NSString \*todate = toD[j];

NSDateFormatter \*toDateFormatter = [[NSDateFormatter alloc]init];

NSTimeZone \*gmtto = [NSTimeZone timeZoneForSecondsFromGMT:19800];

[toDateFormatter setTimeZone:gmtto];

[toDateFormatter setDateFormat:@"yyyy-MM-dd hh:mm:ss"];

NSDate \*toDate = [toDateFormatter dateFromString:todate];

NSComparisonResult from;

NSComparisonResult to;

from = [date compare:fromDate];

to = [date compare:toDate];

if((from == NSOrderedSame) || (to == NSOrderedSame) || (from == NSOrderedDescending && to == NSOrderedAscending)){

NSDictionary \*dict = [[NSDictionary alloc] init];

dict = setFreeHour[j];

\_lat = dict[@"latitude"];

\_longi = dict[@"longitude"];

CLLocationCoordinate2D cordinate;

cordinate.latitude = [\_lat doubleValue];

cordinate.longitude = [\_longi doubleValue];

\_planeLoc = dict[@"location"];

// Add an annotation

MKPointAnnotation \*point1 = [[MKPointAnnotation alloc] init];

point1.coordinate = CLLocationCoordinate2DMake(cordinate.latitude, cordinate.longitude);

point1.title = \_planeName;

point1.subtitle = [[[@"Passenger Cap. : " stringByAppendingString:\_planePassenger] stringByAppendingString:@" @"] stringByAppendingString:\_planeLoc];

[self.mapView addAnnotation:point1];

}

else{

NSLog(@"Not added");

}

}

}

}

// MARK: Map didUpdate Method

- (void)mapView:(MKMapView \*)mapView didUpdateUserLocation:(MKUserLocation \*)userLocation

{

if(timesUpdated<2){

MKCoordinateRegion region = MKCoordinateRegionMakeWithDistance(userLocation.coordinate, 700000, 700000);

[self.mapView setRegion:[self.mapView regionThatFits:region] animated:YES];

[\_locationManager stopUpdatingLocation];

timesUpdated++;

}

}

————————————————————————————————————————

// MARK: Map viewForAnnotation Method - for 2 different annotation

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

if ([annotation isKindOfClass:[MKPointAnnotation class]]){

MKAnnotationView \*pinView = (MKAnnotationView\*)[mapView dequeueReusableAnnotationViewWithIdentifier:@"CustomPinAnnotationView"];

if(!pinView){

pinView = [[MKAnnotationView alloc] initWithAnnotation:annotation reuseIdentifier:@"CustomPinAnnotationView"];

pinView.canShowCallout = YES;

pinView.image = [UIImage imageNamed:@"annotation"];

pinView.calloutOffset = CGPointMake(0, 0);

}

else{

pinView.annotation = annotation;

if ([annotation isKindOfClass:[SecondAnnotation class]]) {

pinView.canShowCallout = YES;

pinView.image = [UIImage imageNamed:@"locationMarker"];

pinView.calloutOffset = CGPointMake(0, 0);

}

else{

pinView.canShowCallout = YES;

pinView.image = [UIImage imageNamed:@"annotation"];

pinView.calloutOffset = CGPointMake(0, 0);

\_hideTable.hidden = YES;

}

}

return pinView;

}

return nil;

}

————————————————————————————————————————

// MARK: Map viewForAnnotation Method - for 1 annotation

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

if ([annotation isKindOfClass:[MKUserLocation class]])

return nil;

if ([annotation isKindOfClass:[MKPointAnnotation class]])

{

MKAnnotationView \*pinView = (MKAnnotationView\*)[mapView dequeueReusableAnnotationViewWithIdentifier:@"CustomPinAnnotationView"];

if (!pinView)

{

pinView = [[MKAnnotationView alloc] initWithAnnotation:annotation reuseIdentifier:@"CustomPinAnnotationView"];

pinView.canShowCallout = YES;

pinView.image = [UIImage imageNamed:@"annotation"];

pinView.calloutOffset = CGPointMake(0, 0);

UIImageView \*iconView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"annotation"]];

pinView.leftCalloutAccessoryView = iconView;

} else {

pinView.annotation = annotation;

}

return pinView;

}

return nil;

}

————————————————————————————————————————

- (void)mapView:(MKMapView \*)mapView didSelectAnnotationView:(MKAnnotationView \*)view{

double getLat;

double getLong;

getLat = view.annotation.coordinate.latitude;

getLong = view.annotation.coordinate.longitude;

NSString \*getNewLat = [NSString stringWithFormat:@"%lf",getLat];

NSString \*getNewLong = [NSString stringWithFormat:@"%lf",getLong];

NSString \*url = [NSString stringWithFormat:@"URL"];

NSData \*data = [NSData dataWithContentsOfURL:[NSURL URLWithString:url]];

NSArray \*array = [NSJSONSerialization JSONObjectWithData:data options:0 error:nil];

\_setFreeHourArray = [[array valueForKey:@"result"] valueForKey:@"set\_free\_hour"];

NSPredicate \*predicate = [NSPredicate predicateWithFormat:@"latitude CONTAINS[c] %@ AND longitude CONTAINS[c] %@",getNewLat,getNewLong];

\_foundPlane = [\_setFreeHourArray filteredArrayUsingPredicate:predicate];

for (int z=0; z<\_foundPlane.count; z++) {

for (int y=0; y<[\_foundPlane[z] count]; y++) {

if([[\_foundPlane[z][y] valueForKey:@"latitude"] isEqualToString:getNewLat]){

[\_display addObject:\_foundPlane[z][y]];

}

else{

NSLog(@"LAT is not same, Not Added");

}

}

}

}