```
//
//  SQLDatabase.m
//  SFHS News
//
//  Created by Harish Kamath on 1/4/15.
//  Copyright (c) 2015 Harish Kamath. All rights reserved.
//

#import "SQLDatabase.h"
#import "EventInfo.h"
#import "ClubInfo.h"

@implementation SQLDatabase
@synthesize errorBool;

static SQLDatabase *_database;

+(SQLDatabase *)database{
    if(_database == nil)
        _database = [[SQLDatabase alloc] init];

    return _database;

}

//This method is what creates the connection to the remote PHP Script(which then
    queries the database) gathers the information, and stores it in our local
    database variable.
-(id)init{
    NSLog(@"SQL Database Initialized!");
    dictArrayEvents = [[NSMutableArray alloc] init];
    dictArrayClubs = [[NSMutableArray alloc] init];
    if(self = [super init]){
        NSURL *url = [NSURL URLWithString:@"http://www.sfhsnews.com/News/
            eventPull.php"];
        NSURLRequest *urlRequest = [NSURLRequest requestWithURL:url];

        NSURL *urlClub = [NSURL URLWithString:@"http://www.sfhsnews.com/News/
            clubPull.php"];
        NSURLRequest *urlRequestClub = [NSURLRequest requestWithURL:urlClub];


        [NSURLConnection sendAsynchronousRequest:urlRequest queue:
            [NSOperationQueue mainQueue] completionHandler:^(NSURLResponse *
            response, NSData *data, NSError *connectionError){
            if(connectionError){
                NSLog(@"%s: sendAsynchronousRequest error:%@", __FUNCTION__,
                    connectionError);
                errorBool = YES;
            return;
            }

           // NSLog(@"raw JSON: %@", [[NSString alloc] initWithData:data
               encoding:NSUTF8StringEncoding]);
            NSError *parseError = nil;
            NSArray *dataArray = [[[NSString alloc] initWithData:data encoding:
```

```
            NSUTF8StringEncoding] componentsSeparatedByString:@"}"];
        for(NSString *string in dataArray){
            if (string.length > 1) {


            NSString *finishedString = [NSString stringWithFormat:@"%@}",
                string];
            NSData *tempData = [finishedString dataUsingEncoding:
                NSUTF8StringEncoding];
           // NSLog(@"Data for string\"%@\" is %@", finishedString,
               tempData);
                NSDictionary *temp = [NSJSONSerialization JSONObjectWithData:
                    tempData options:kNilOptions error:&parseError];
                NSDictionary *results = [self
                    nullFreeDictionaryWithDictionary:temp] ;

        if (parseError) {
            NSLog(@"Parse Error for Events! %@", parseError);
            return;
        }
                [dictArrayEvents addObject:results];
            }

        }

    }];




[NSURLConnection sendAsynchronousRequest:urlRequestClub queue:
    [NSOperationQueue mainQueue] completionHandler:^(NSURLResponse *response,
    NSData *data, NSError *connectionError){
    if(connectionError){
        NSLog(@"%s: sendAsynchronousRequest error:%@", __FUNCTION__,
            connectionError);
        return;
    }

    // NSLog(@"raw JSON: %@", [[NSString alloc] initWithData:data
        encoding:NSUTF8StringEncoding]);
    NSError *parseError = nil;
    NSArray *dataArray = [[[NSString alloc] initWithData:data encoding:
        NSUTF8StringEncoding] componentsSeparatedByString:@"}"];

    for(NSString *string in dataArray){
        if (string.length > 1) {


            NSString *finishedString = [NSString stringWithFormat:@"%@}",
                string];
            NSData *tempData = [finishedString dataUsingEncoding:
                NSUTF8StringEncoding];
            // NSLog(@"Data for string\"%@\" is %@", finishedString,
                tempData);
            NSDictionary *temp = [NSJSONSerialization JSONObjectWithData:
```

```
                    tempData options:kNilOptions error:&parseError];
                NSDictionary *results = [self nullFreeDictionaryWithDictionary:
                    temp];
                if (parseError) {
                    NSLog(@"Parse Error for Clubs! %@", parseError);
                    return;
                }
                [dictArrayClubs addObject:results];
            }

        }

    }];

    }
    return self;
}

-(NSArray *)eventInfos{
   // NSLog(@"eventInfos of SQL Database created!");
    NSMutableArray *retVals = [[NSMutableArray alloc] init];
    for(NSDictionary *dict in dictArrayEvents){

        EventInfo *info = [[EventInfo alloc] initWithUniqueId:(int)[[dict
            objectForKey:@"uniqueID"] integerValue] name:[dict
            objectForKey:@"name"] desc:[dict objectForKey:@"desc"] founder:[dict
            objectForKey:@"founder"] timeStart:[dict objectForKey:@"timeStart"]
            date:[dict objectForKey:@"date"] timeEnd:[dict
            objectForKey:@"timeEnd"] address:[dict objectForKey:@"address"] club:
            [dict objectForKey:@"club"] results:[dict objectForKey:@"results"]
            imageLink:[dict objectForKey:@"imageLink"] extraInfo:[dict
            objectForKey:@"extraInfo"]];
        [retVals addObject:info];
    }
    return retVals;

}

-(NSArray *)clubInfos{
    //NSLog(@"clubInfos of SQL Database created!");
    NSMutableArray *retVals = [[NSMutableArray alloc] init];
    for(NSDictionary *dict in dictArrayClubs){
        ClubInfo *info = [[ClubInfo alloc] initWithUniqueId:(int)[[dict
            objectForKey:@"uniqueID"] integerValue] name:[dict
            objectForKey:@"name"] desc:[dict objectForKey:@"desc"] advisor:[dict
            objectForKey:@"advisor"] meeting:[dict objectForKey:@"meetings"]
            contact:[dict objectForKey:@"contact"] events:[dict
            objectForKey:@"events"] results:[dict objectForKey:@"results"]
            imageLink:[dict objectForKey:@"picture"] facebook:[dict
            objectForKey:@"facebook"] twitter:[dict objectForKey:@"twitter"]];
        [retVals addObject:info];
    }
    return retVals;

}
```

```
- (NSDictionary *)nullFreeDictionaryWithDictionary:(NSDictionary *)dictionary
{
    NSMutableDictionary *replaced = [NSMutableDictionary
        dictionaryWithDictionary:dictionary];
    // Iterate through each key-object pair.
    [dictionary enumerateKeysAndObjectsUsingBlock:^(id key, id object, BOOL *stop
        ) {
        // If object is a dictionary, recursively remove NSNull from dictionary.
        if ([object isKindOfClass:[NSDictionary class]]) {
            NSDictionary *innerDict = object;
            replaced[key] = [self nullFreeDictionaryWithDictionary:innerDict];
        }
        // If object is an array, enumerate through array.
        else if ([object isKindOfClass:[NSArray class]]) {
            NSMutableArray *nullFreeRecords = [NSMutableArray array];
            for (id record in object) {
                // If object is a dictionary, recursively remove NSNull from
                    dictionary.
                if ([record isKindOfClass:[NSDictionary class]]) {
                    NSDictionary *nullFreeRecord = [self
                        nullFreeDictionaryWithDictionary:record];
                    [nullFreeRecords addObject:nullFreeRecord];
                }
                else {
                    if (object == [NSNull null]) {
                        [nullFreeRecords addObject:@""];
                    }
                    else {
                        [nullFreeRecords addObject:record];
                    }
                }
            }
            replaced[key] = nullFreeRecords;
        }
        else {
            // Replace [NSNull null] with nil string "" to avoid having to
                perform null comparisons while parsing.
            if (object == [NSNull null]) {
                replaced[key] = @"";
            }
        }
    }];

    return [NSDictionary dictionaryWithDictionary:replaced];
}


@end
```