```swift
//
//  AdminBeaconViewController.swift
//  AttendanceApplication
//


import Foundation
import UIKit
import Alamofire

class AdminBeaconTabViewController: UIViewController {

    // MARK: IBOutlets

    @IBOutlet weak var beaconTableView: UITableView!

    // MARK: Private Properties

    private var searchController = UISearchController(searchResultsController: nil)
    private var beacons: [Beacon] = []
    private var filteredBeacons: [Beacon] = [Beacon]()

    // MARK: UIViewController Methods

    override func viewDidLoad() {
        super.viewDidLoad()

        self.beaconTableView.delegate = self
        self.beaconTableView.dataSource = self

        searchController.searchResultsUpdater = self
        searchController.dimsBackgroundDuringPresentation = false
        definesPresentationContext = true
        beaconTableView.tableHeaderView = searchController.searchBar
    }

    override func viewWillAppear(_ animated: Bool){
        getBeacons()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Called before segue is performed. Used to pass the beacon object selected
         to the detail view.
        if segue.identifier == "beaconTableToDetail" {
            let destination_VC = segue.destination as!
             AdminBeaconDetailViewController
            destination_VC.beacon = sender as? Beacon
        }
    }

    // MARK: Internal Methods

    internal func getBeacons(){
        let parameters: Parameters = [
```

```
            "type": "admin.get_beacons",
            "args": [
                "query": ""
            ]
        ]

        Alamofire.request(HTTPHelper.url, method: .post, parameters: parameters,
         encoding: JSONEncoding.default).responseJSON {
            response in

            switch response.result {
            case .failure( _):

                return

            case .success(let data):
                // First make sure a dictionary is recieved: Data validation
                guard let json = data as? [String : AnyObject] else {
                    // Print statement for debugging purposes, not seen by users.
                    print("Failed to get expected dictionary from webserver.")
                    return
                }

                // Then make sure that key/value pairs are correct: Data validation
                guard let success = json["successful"] as? Int, let beacons =
                 json["beacons"] as? [[String: String]] else {
                    // Print statement for debugging purposes, not seen by users.
                    print("Failed to get expected data from webserver")
                    return
                }

                if success == 1 {
                    var beaconObjectList = [Beacon]()
                    for beacon in beacons {
                        beaconObjectList.append(Beacon(beaconJSONObject: beacon))
                    }
                    self.beacons = beaconObjectList
                    self.beaconTableView.reloadData()

                } else {
                }
            }
        }
    }
}

// MARK: Extentions

extension AdminBeaconTabViewController: UISearchResultsUpdating {

    func updateSearchResults(for searchController: UISearchController) {
        filteredBeacons = beacons.filter({ (beacon: Beacon) -> Bool in
            if beacon.roomNumber.contains(searchController.searchBar.text!) {
                return true
```

```
            } else {
                return false
            }
        })
        self.beaconTableView.reloadData()
    }

}

extension AdminBeaconTabViewController: UITableViewDataSource, UITableViewDelegate {

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) ->
     Int {
        /*
         * Function called by the program to check how many students exist in the
          students array, and therefore how many StudentCells are necessary.
         */
        if searchController.isActive && searchController.searchBar.text != "" {
            return filteredBeacons.count
        }

        return beacons.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
     UITableViewCell {
        /*
         * Function called at the creation of every new cell in the table. It takes
          the prototype cell (casted to a StudentCell) and adds the relevant labels.
         */

        let beacon: Beacon

        if searchController.isActive && searchController.searchBar.text != "" {
            beacon = filteredBeacons[filteredBeacons.count - indexPath.row - 1]
        } else {
            beacon = beacons[beacons.count  - indexPath.row - 1]
        }

        let cell = tableView.dequeueReusableCell(withIdentifier: "BeaconCell") as!
         BeaconCell
        cell.setLabels(beacon: beacon)
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        /*
         * Function called when a item is selected. Performs the segue to the detail
          view.
         */

        let beacon: Beacon

        if searchController.isActive && searchController.searchBar.text != "" {
```

```
            beacon = filteredBeacons[filteredBeacons.count - indexPath.row - 1]
        } else {
            beacon = beacons[beacons.count  - indexPath.row - 1]
        }

        performSegue(withIdentifier: "beaconTableToDetail", sender: beacon)
    }
}
```