

```

//
// AdminStudentTabViewController.swift
// AttendanceApplication
//

import Foundation
import UIKit
import Alamofire

class AdminStudentTabViewController: UIViewController {

    // MARK: IBOutlets

    @IBOutlet weak var studentTableView: UITableView!

    // MARK: Private Properties

    private var searchController = UISearchController(searchResultsController: nil)
    private var students: [Student] = []
    private var filteredStudents: [Student] = [Student]()

    // MARK: UIViewController methods

    override func viewDidLoad() {
        super.viewDidLoad()

        // Set the delegate and the data source for the table
        studentTableView.delegate = self
        studentTableView.dataSource = self

        // Setup the search controller
        searchController.searchResultsUpdater = self
        searchController.dimsBackgroundDuringPresentation = false
        definesPresentationContext = true
        studentTableView.tableHeaderView = searchController.searchBar

        // Get the students and update the table
        getStudents()
    }

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

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "studentTableToDetail" {
            let destination_VC = segue.destination as!
                AdminStudentDetailViewController
            destination_VC.student = sender as? Student
        }
    }

    // MARK: Internal methods

```

```

internal func getStudents(){
    let parameters: Parameters = [
        "type": "admin.get_students",
        "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
            }

            // Check if user is logged in according to the server. If not
            // redirect user to login page.
            if (json["login_necessary"] as? Int) != nil {
                // Pass off controls to the login view controller by performing
                // a logout.
                return
            }

            // Then make sure that key/value pairs are correct: Data validation
            guard let success = json["successful"] as? Int, let students =
                json["students"] 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 studentObjectList = [Student]()
                for student in students {
                    studentObjectList.append(Student(studentJSONObject:
                        student))
                }
                self.students = studentObjectList
                self.studentTableView.reloadData()

            } else {
            }
        }
    }
}

```

```

    }
}

// MARK: Extensions

extension AdminStudentTabViewController: UISearchResultsUpdating {

    func updateSearchResults(for searchController: UISearchController) {
        filteredStudents = students.filter({ (student: Student) -> Bool in
            if student.name.contains(searchController.searchBar.text!) {
                return true
            } else {
                return false
            }
        })
        self.studentTableView.reloadData()
    }

}

extension AdminStudentTabViewController: 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 filteredStudents.count
        }

        return students.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 student: Student

        if searchController.isActive && searchController.searchBar.text != "" {
            student = filteredStudents[indexPath.row]
        } else {
            student = students[indexPath.row]
        }

        let cell = tableView.dequeueReusableCell(withIdentifier: "StudentCell") as!
        StudentCell
        cell.setLabels(student: student)
    }
}

```

```
        return cell
    }

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

        let student: Student

        if searchController.isActive && searchController.searchBar.text != "" {
            student = filteredStudents[indexPath.row]
        } else {
            student = students[indexPath.row]
        }

        performSegue(withIdentifier: "studentTableToDetail", sender: student)
    }
}
```