```swift
//
//  AdminStudentDetailViewController.swift
//  AttendanceApplication
//

import UIKit
import Alamofire

class AdminStudentDetailViewController: UIViewController {

    // MARK: IBOutlets

    @IBOutlet weak var studentEmail: UILabel!
    @IBOutlet weak var studentName: UILabel!
    @IBOutlet weak var locationTableView: UITableView!

    // MARK: Private properties

    private var locations: [Location] = []

    // MARK: Public properties

    var student: Student!

    // MARK: UIViewController methods

    override func viewDidLoad() {
        super.viewDidLoad()

        self.locationTableView.delegate = self
        self.locationTableView.dataSource = self

        // Get the labels and locations and update the labels and table
        setLabels()
        getLocations()
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "locationTableToDetail" {
            let destination_VC = segue.destination as! MapViewController
            destination_VC.location = sender as? Location
        }
    }

    // MARK: Internal methods

    internal func getLocations(){
        /*
         * Function that retrieves all locations in the past day and returns them as
           a dynamic array of Locations.
         */
        let parameters: Parameters = [
            "type": "admin.get_student_location",
            "args": [
```

```swift
                    "student_id": self.student.id
                ]
            ]

            Alamofire.request(HTTPHelper.url, method: .post, parameters: parameters,
             encoding: JSONEncoding.default).responseJSON { (response) in
                switch response.result {
                case .failure( _):
                    print("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 locations =
                     json["results"] 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 locationObjectList = [Location]()

                        for location in locations {
                            locationObjectList.append(Location(locationJSONObject:
                             location))
                        }
                        self.locations = locationObjectList
                        self.locationTableView.reloadData()
                    } else {

                    }
                }
            }
        }
    }

    internal func setLabels() {
        self.studentEmail.text = student.email
        self.studentName.text = student.name
    }
}

// MARK: Extentions

extension AdminStudentDetailViewController: UITableViewDataSource,
 UITableViewDelegate {
```

```swift
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) ->
     Int {
        return locations.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
     UITableViewCell {
        let location = locations[locations.count - indexPath.row - 1]
        let cell = locationTableView.dequeueReusableCell(withIdentifier:
         "LocationCell") as! LocationCell
        cell.setLabels(location: location)
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        let location: Location
        location = locations[locations.count - indexPath.row - 1]
        performSegue(withIdentifier: "locationTableToDetail", sender: location)
    }

}
```