|  |  |
| --- | --- |
| **Student Identification** |  |
| Name: Somiya Tyagi\_ Date:  Student number: Group number: Result: \_   * I declare that this is an original work, and that I credited all content sources of which I am   not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work. | |

|  |  |  |
| --- | --- | --- |
| □ **#1** | 10 pts | No messages presented to the user |

* No messages or dialogs were sent to the user. The application is only printing messages at XCode console. You have to replace those messages using a better way to notify the user.

**Answer:**

if studentName.count < 3 {

print("Please, enter a student name with at least 3 chars!")

self.alert(message: "Please, enter a student name with at least 3 chars!")

return

}

print("Enter a valid name!")

if name.count > 7 || name.count > 8 {

print("Please, enter a name with at least 7 or 8 chars!")

self.alert(message: "Please, enter a name with at least 7 or 8 chars!")

return false

}

|  |  |  |
| --- | --- | --- |
| **#2** | 10 pts | Login page with user and password not implemented |

* The access should be validated by username and password combination.
* The username validation is not well implemented. The user must enter a value with 7 or 8 digits.
* The correct username to access the application should be **your student number**.
* The password should be masked (cannot show while typing) and the expected value is “**adm123**”.

class ViewController: UIViewController {

@IBOutlet weak var txtName : UITextField!

@IBOutlet weak var txtPswd: UITextField!

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

}

override func shouldPerformSegue(withIdentifier identifier: String, sender: Any?) -> Bool {

if identifier == Segue.toStudentList {

guard let name : String = txtName.text else {

print("Please, enter a valid name!")

self.alert(message: "Please, enter a valid name!")

return false

}

if name.count > 7 || name.count > 8 {

print("Please, enter a name with at least 7 or 8 chars!")

self.alert(message: "Please, enter a name with at least 7 or 8 chars!")

return false

}

if self.txtName.text != "2111849" && self.txtPswd.text != "adm123" {

return false

}

return true

}

return false

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

if segue.identifier == Segue.toStudentList {

let studentList = (segue.destination as! StudentListViewController)

studentList.userName = self.txtName.text!

}

}

}

|  |  |  |
| --- | --- | --- |
| □ **#3** | 5 pts | Logged username not presented on StudentListViewController |

The username informed at ViewController is not being presented on StudentListViewController. The label should be filled with the text “**Hello**“ followed by the user login (e.g.: “**Hello 1234567**”)

**Answer:**

// Do any additional setup after loading the view.

lblUsername.text = "Hello" + " \(userName)"

func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return StudentProvider.allStudents.count

}

cell.textLabel?.text = StudentProvider.allStudents[indexPath.row].getName()

if segue.identifier == Segue.toStudentInfoEditing {

..

}

|  |  |  |
| --- | --- | --- |
| □ **#4** | 20 pts | StudentListViewController: TableView is not working |

* The table view should list all students, showing the student’s name on each row.

**Answer:**

class StudentListViewController: UIViewController, UITableViewDelegate, UITableViewDataSource, TableViewRefresh {

public var userName : String = ""

private var selectedRow : Int = -1

@IBOutlet weak var lblUsername : UILabel!

@IBOutlet weak var tableView: UITableView!

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

lblUsername.text = "Hello" + " \(userName)"

tableView.delegate = self

tableView.dataSource = self

}

func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return StudentProvider.allStudents.count

}

func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)

cell.textLabel?.text = StudentProvider.allStudents[indexPath.row].getName()

return cell

}

func tableView(\_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

self.selectedRow = indexPath.row

tableView.deselectRow(at: indexPath, animated: false)

performSegue(withIdentifier: Segue.toStudentInfoEditing, sender: nil)

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

if segue.identifier == Segue.toStudentInfo {

/// insert operation

let studentInfo = (segue.destination as! StudentInfoViewController)

studentInfo.delegate = self /// deletage to auto-refresh the list (don't touch this)

studentInfo.editMode = false

return

}

if segue.identifier == Segue.toStudentInfoEditing {

/// Update/delete operations

let studentInfo = (segue.destination as! StudentInfoViewController)

studentInfo.delegate = self /// deletage to auto-refresh the list (don't touch this)

studentInfo.editMode = true

studentInfo.selectedStudent = StudentProvider.allStudents[selectedRow]

}

}

func refresh() {

tableView.reloadData()

} }

|  |  |  |
| --- | --- | --- |
| □ **#5** | 5 pts | StudentListViewController: Title incorrect |

The title is incompatible with the operation type being performed. It should be “**Adding new Student**” when *editMode == False* and **“Showing Student**” when *editMode == True*.

**Answer:**

override func viewDidLoad() {

super.viewDidLoad()

if editMode {

title = "Editing student"

btnDelete.isHidden = false

txtStudentName.text = selectedStudent!.getName()

txtStudentEmail.text = selectedStudent!.getEmail()

} else {

title = "Adding new student"

btnDelete.isHidden = true

}

}

|  |  |  |
| --- | --- | --- |
| □ **#6** | 10 pts | StudentInfoViewController: Delete button missing |

* The delete button is always invisible. It should be visible when the user reaches this view controller coming from a previous selection on StudentListViewController TableView.

**Answer:**

@IBOutlet weak var btnDelete : UIButton!

override func viewDidLoad() {

super.viewDidLoad()

if editMode {

title = "Editing student"

btnDelete.isHidden = false

txtStudentName.text = selectedStudent!.getName()

txtStudentEmail.text = selectedStudent!.getEmail()

} else {

title = "Adding new student"

btnDelete.isHidden = true

}

}

@IBAction func btnDeleteTouchUp(\_ sender: Any) {

let alert = UIAlertController(title: "Message", message: "Do you really want to delete " + "\(self.txtStudentName.text!) ?", preferredStyle: .alert)

alert.addAction(UIAlertAction(title: "OK", style: .default, handler: { [self] action in

StudentProvider.removeStudent(studentId: selectedStudent!.getId())

self.delegate?.refresh()

self.navigationController?.popViewController(animated: true)

}))

alert.addAction(UIAlertAction(title: "Cancel", style: .default, handler: { action in

}))

self.present(alert, animated: true, completion: nil)

|  |  |  |
| --- | --- | --- |
| □ **#7** | 10 pts | StudentInfoViewController: Student information is not being presented |

* Student information (name and email) is not presented when the user is redirected to this controller after selecting a student on StudentListViewController TableView.

**Answer:**

@IBOutlet weak var txtStudentEmail: UITextField!

if editMode {

selectedStudent!.setName(name: studentName)

selectedStudent!.setEmail(email: self.txtStudentEmail.text!)

} else {

let student = Student()

student.setName(name: studentName)

student.setEmail(email: self.txtStudentEmail.text!)

student.setId(id: Student.getNextId())

StudentProvider.addStudent(student: student)

}

|  |  |  |
| --- | --- | --- |
| □ **#8** | 10 pts | StudentInfoViewController: User cannot inform an email address for a student |

* Field email is missing on StudentInfoViewController UI. You have to implement it.
* Email information should be saved into a student object. You have to implement it.

**Answer:**

else {

print("Enter a valid name!")

self.alert(message: "Enter a valid name!")

}

if !isValidEmail(self.txtStudentEmail.text!) {

self.alert(message: "Please, enter a valid email address!")

return

}

extension StudentInfoViewController

{

func isValidEmail(\_ email: String) -> Bool {

let emailRegEx = "[A-Z0-9a-z.\_%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,64}"

let emailPred = NSPredicate(format:"SELF MATCHES %@", emailRegEx)

return emailPred.evaluate(with: email)

}

}

|  |  |  |
| --- | --- | --- |
| □ **#9** | 10 pts | StudentInfoViewController: StudentProvider.updateStudent method not working |

* There is an issue when updating student information. Review this method at StudentProvider Class.

**Answer:**

class StudentProvider {

static var allStudents : [Student] = [Student(id: 1, name: "AMAN", email: "Aman@gmail.com"),Student(id: 2, name: "BOMAN", email: "boman@gmail.com")]

static func addStudent( student : Student ){ .. }

|  |  |  |
| --- | --- | --- |
| □ **#10** | 10 pts | StudentInfoViewController: There is no delete confirmation |

* You should present a confirmation message before deleting a student: “**Do you really want to delete *[STUDENT NAME]***? “ (e.g.: “Do you really want to delete Daniel?” ).

**Answer:**

let alert = UIAlertController(title: "Message", message: "Do you really want to delete " + "\(self.txtStudentName.text!) ?", preferredStyle: .alert)

alert.addAction(UIAlertAction(title: "OK", style: .default, handler: { [self] action in

StudentProvider.removeStudent(studentId: selectedStudent!.getId())

self.delegate?.refresh()

self.navigationController?.popViewController(animated: true)

}))

alert.addAction(UIAlertAction(title: "Cancel", style: .default, handler: { action in

}))

self.present(alert, animated: true, completion: nil)