REVISION EXERCISE

1. Problem Question

The Storyboard (full) is given in Appendix A

PART 1

(a) Figure 1(a) shows the UI of the `LoginViewController` in the Main Storyboard. Create a complete `LoginViewController` that met the following criteria:

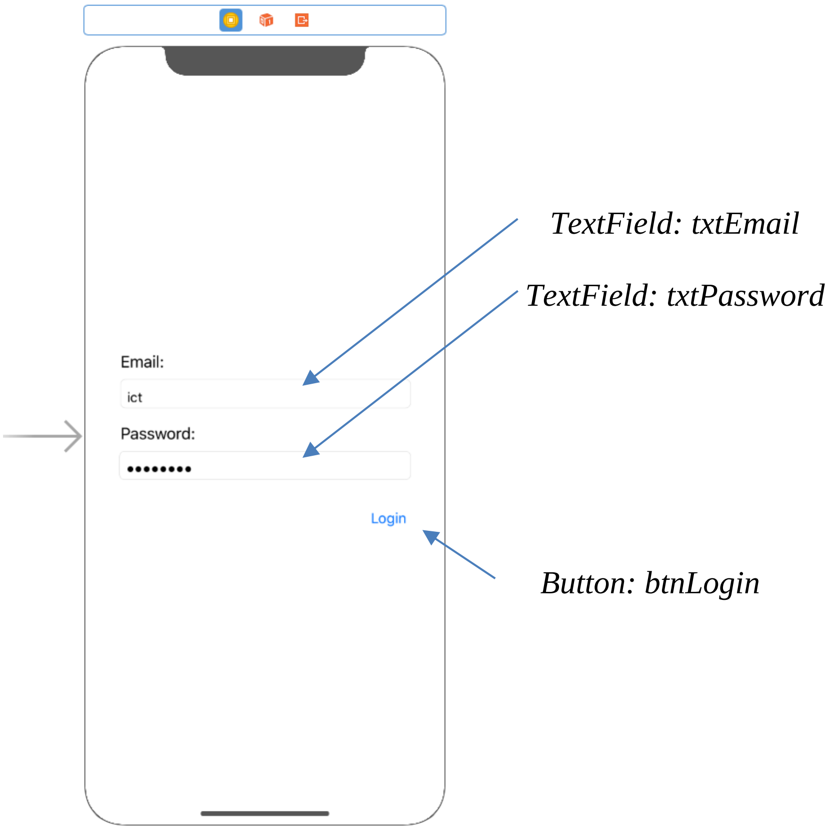
i. A new `LoginViewController` Class using Swift programming language

ii. Populate the screen with 2 `TextFields`, 2 `Labels` and a `Button`. (Note: Configure all the properties appropriately)

iii. Link the 2 `TextField` to the `LoginViewController` Class

iv. Set as an initial View Controller

(15 marks)



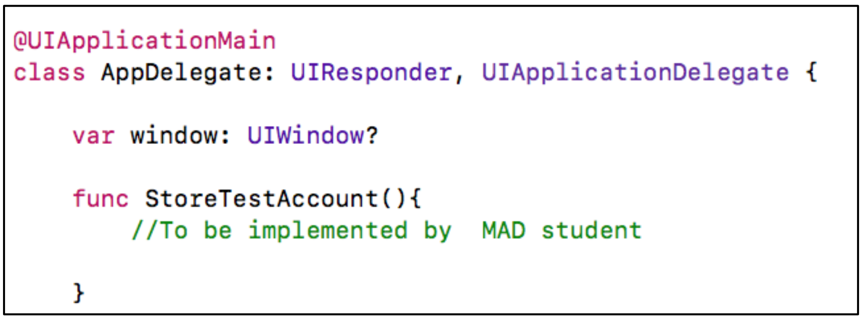
(b) Design and Create the **`Account`** Entity Data Class in the `xcdatamodeld` file. The attributes are given in the following table.

Attributes	**Type**
email	String
Password	String

(5 marks)

(c) Implement the `StoreTestAccount` function in the `AppDelegate` Class. This function stores the following data if the `Account` Core Data is empty. Figure 1(c) shows the `StoreTestAccount` function.

Email	Password
user1@gmail.com	12345678
admin@apple.com	root



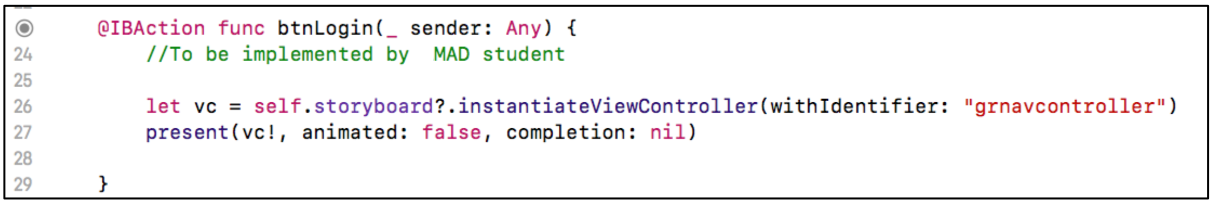
```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func StoreTestAccount(){
        //To be implemented by  MAD student

    }
```

Figure 1(c) `StoreTestAccount` function

(15 marks)

(d) Once the Login button is clicked, check if the user is a valid user from the Core Data, if valid, present the `GRTableViewController` on the screen An example of the incomplete login button function is shown below.

(10 marks)



```
    @IBAction func btnLogin(_ sender: Any) {
        //To be implemented by  MAD student

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "grnavcontroller")
        present(vc!, animated: false, completion: nil)

    }
```

PART 2

(a) Given the Class Element Diagram, create a new Contact Class in the Xocde project.

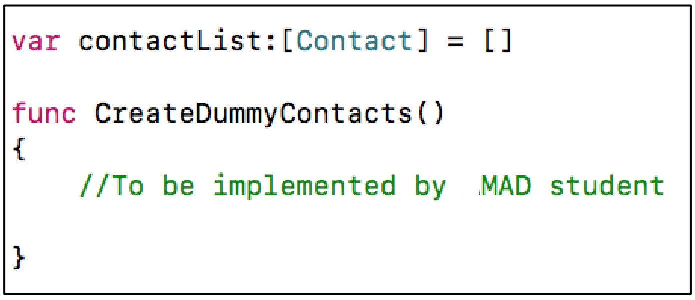
Contact
-firstname:String -lastname:String -photo:UIImage -telephone:String
+init(String, String, UIImage, String)

(5 marks)

(b) Implement the `CreateDummyContacts` function in the `AppDelegate` Class. 2 contacts into the `contactList`. The Figure 2(b) shows the `CreateDummyContacts` function.

(10 marks)

firstname	**lastname**	**photo**	**telephone**
Alan	Hayes	UIImage(named: "alanhayes")!	88221122
Jean	Yip	UIImage(named: "jeanyip")!	55115522



```
var contactList:[Contact] = []

func CreateDummyContacts()
{
    //To be implemented by  MAD student

}
```

Figure 2(b)

(c) Using the information in 2(b), complete the TableView functions `GRUITableViewController` Class with the following criteria:

i. Override the `numberOfSections` function with 1 section only.

ii. Override the `TableView` with the “`numberOfRowsInSection`” signature accordingly.

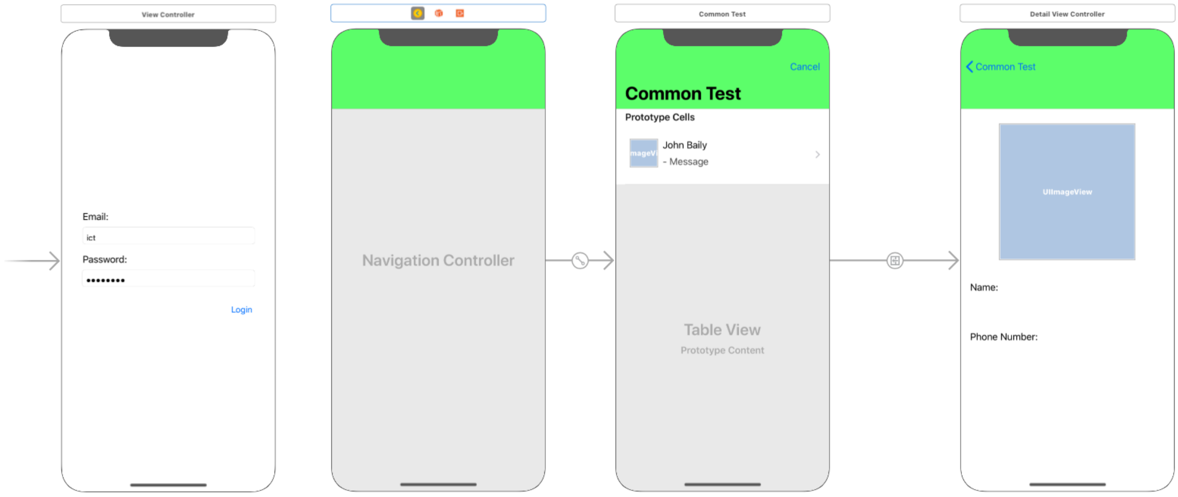
iii. Override the `TableView` with the “`cellForRowAt`” signature accordingly.

(20 marks)

Note: The `GRTableViewCell` Class has already been in the `GRUITableViewController.swift` file and the two variables (`imgPhoto` and `txtName`) has been linked to the Main Storyboard.

Appendix A

to be implemented only


View Controller
Email:
ict
Password:
••••••••
Login
Navigation Controller
Common Test
Cancel
Common Test
Prototype Cells
John Baily
- Message
Table View
Prototype Content
Detail View Controller
Common Test
UIImageView
Name:
Phone Number: