// check using Core Data in the begining
// design software -> Sketch
// Art Book Project.xcdatamodeld -> Add Entity => change the Entity name

Entities	Attribute	
paintings	artist	String
	id	UUID // universal unique ID
	image	Binary Data
	name	String
	year	Integer 32

// editor -> embeded in -> navigation controller
Yellow color -> 1-0-1 -> update constraints

// bug fix -> open storyboard as source code => find the missing <viewController id customClass customModule ---> produce clean -> product build

// test => ViewController.swift

```
override func viewDidLoad(){
    super.viewDidLoad
    navigationController.navigationBar.topItem.rightBarButtonItem = UIBarButtonItem(barButtonSystemItem:
        UIBarButtonItem.SystemItem.add, target: self, action(addButtonClicked))
}
@objc func addButtonClicked(){
    performSegue(withIdentifier: "toDetailsVC", sender: nil)
}
```

// DetailsVC.swift

```
import UIKit
import CoreData
```

<- important to use the image picker

```
class DetailsVC: UIViewController, UIImagePickerControllerDelegate, UINavigationControllerDelegate {
    @IBOut let weak var imageView: UIImageView!
    | Similar ones
    @IBOutlet weak var saveButton: UIButton! // not only action but also outlet to reach attributes
    var chosenPainting = ""
    var chosenPaintingId: UUID?
    override func viewDidLoad(){
        super.viewDidLoad()
        if chosenPainting != "{
            saveButton.isHidden = true

            let appDelegate = UIApplication.shared.delegates as! AppDelegate
            let context = appDelegate.persistentContainer.viewContext
            let fetchRequest = NSFetchRequest<NSFetchRequestResult>(entityName: "Paintings")
            let idString = chosenPaintingId?.uuidString
            fetchRequest.predicate = NSPredicate(format: "id = %@", idString!)
                                                  -> regular expression, from C => put idString! here
            fetchRequest.returnsObjectsAsFaults = false

            do{
                let results = try context.fetch(fetchRequest)
                if result.count > 0{
                    for result in results as! [NSManagedObject]{
                        if let name = result.value(forKey: "name") as? String{
```

```
                nameText.text = name
            }
            if let artist = result.value(forKey: "artist") as? String {
                artistText.text = artist
            }
            if let year = result.value(forKey: "year") as? Int {
                yearText.text = String(year)
            }
            if let imageData = result.value(forKey: "image") as? Data {
                let image = UIImage(data: imageData)
                imageView.image = image
            }
        }
    }
} else {
    saveButton.isHidden = false
    saveButton.isEnabled = false
    nameText.text = ""
    artistText.text = ""
    yearText.text = ""
}
//Recognizers
let gestureRecognizer = UITapGestureRecognizer(target: self, action: #selector(hideKeyboard))
view.addGestureRecognizer(gestureRecognizer)

imageView.isUserInteractionEnabled = true
let imageTapRecognizer = UITapGestureRecognizer(target: self, action: #selector(selectImage))
imageView.addGestureRecognizer(imageTapRecognizer)
}
@objc func selectImage(){
    let picker = UIImagePickerController()
    picker.delegate = self
    picker.sourceType = .photoLibrary
    picker.allowEditing = true
    present(picker, animated: true, completion: nil)
}

func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info:
    [UIImagePickerController.InfoKey : Any]) {
    imageView.image = info[.originalImage] as? UIImage
    saveButton.isEnabled = true
    self.dismiss(animated: true, completion: nil)
}
//before IOS12 => info.plist => plus button => Privacy - Photo Library Usage Description => Value: to access photos.   ← explaination

@objc func hideKeyboard(){
    view.endEditing(true)
}
@IBAction func saveButtonClicked(_ sender: Any) {
    let appDelegate = UIApplication.shared.delegate as! AppDelegate
    let context = appDelegate.persistentContainer.viewContext
```

```
        let newPainting = NSEntityDescription.insertNewObject(forEntityName: "Paintings", into: context)
        //Attributes
        newPainting.setValue(nameText.text!, forKey: "name")
        newPainting.setValue(artistText.text!, forKey: "artist")
        if let year = Int(yearText.text!){
            newPainting.setValue(year, forKey: "year")
        }
        newPainting.setValue(UUID(), forKey: "id")
        let data = imageView.image.jpegData(compressionQuality: 0.5)
        newPainting.setValue(data, forKey: "image")
        do{
            try context.save() //must wrapped inside a try catch
            print("success")
        }catch{
            print("error")
        }
        NotificationCenter.default.post(name: NSNotification.Name("newData"), object: nil)
        self.navigationController?.popViewController(animated: true) // will not show the new data
    }
}

//ViewController.swift
import UIKit
import CoreData
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource{
    @IBOutlet weak var tableView: UITableView!
    var nameArray = [String]()
    var idArray = [UUID]()
    var selectedPainting = ""
    var selectedPaintingId: UUID?

    override func viewDidLoad(){
        super.viewDidLoad()
        tableView.delegate = self
        tableView.dataSource = self
        getData()
        navigationController.navigationBar.topItem.rightBarButtonItem = UIBarButtonItem(barButtonSystemItem:
            UIBarButtonItem.SystemItem.add, target: self, action(addButtonClicked))
    }

    override func viewWillAppear(_ animated: Bool){
        NotificationCenter.Default.addObserver(self, selector: #selector(getData), name: NSNotification.Name(
            rawValue: "newData"), object: nil)
    }

    @objc func getData(){
        nameArray.removeAll(keepingCapacity: false)
        idArray.removeAll(keepingCapacity: false)
```

```
let appDelegate = UIApplication.shared.delegate as! AppDelegate
let context = appDelegate.persistentContainer.viewContext

let fetchRequest = NSFetchRequest<NSFetchRequestResult>(entityName: "Paintings")
fetchRequest.returnsObjectAsFaults = false
do{
    let results = try context.fetch(fetchRequest)
    if results.count > 0{
        for result in results as![NSManagedObject]{
            if let name = result.value(forKey: "name") as? String{
                self.nameArray.append(name)
            }
            if let id = result.value(forKey: "id") as? UUID{
                self.idArray.append(id)
            }
            self.tableView.reloadData()
        }
    }
}catch{
    print("error")
}
}

@objc func addButtonClicked(){
    selectedPainting = ""

    performSegue(withIdentifier: "toDetailsVC", sender: nil)
}

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

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell{
    let cell = UITableViewCell()
    cell.textLabel?.text = nameArray[indexPath.row]
    return cell
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?){
    if segue.identifier == "toDetailsVC"{
        let destinationVC = segue.destination as! DetailsVC
        destinationVC.chosenPainting = selectedPainting
        destinationVC.chosenPaintingId = selectedPaintingId
    }


}
```

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath){
    selectedPaintImg = nameArray[indexPath.row]
    selectedPaintingId = idArray[indexPath.row]
    performSegue(withIdentifier: "toDetailsVC", sender: nil)
}

func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath:
    IndexPath){
    if editingStyle == .delete{
        let appDelegate = UIApplication.shared.delegate as! AppDelegate
        let context = appDelegate.persistentContainer.viewContext
        let fetchRequest = NSFetchRequest<NSFetchRequestResult>(entityName: "Paintings")
        let idString = idArray[indexPath.row].uuidString
        fetchRequest.predicate = NSPredicate(format: "id = %@", idString)
        fetchRequest.returnObjectsAsFaults = false
        do{
            let results = try context.fetch(fetchRequest)
            if results.count > 0{
                for result in results as! [NSManagedObject]{
                    if let id = result.value(forKey: "id") as? UUID{
                        if id == idArray[indexPath.row]{
                            context.delete(result)
                            nameArray.remove(at: indexPath.row)
                            idArray.remove(at: indexPath.row)
                            self.tableView.reloadData()
                            do{
                                try context.save()
                            }catch{
                                print("error")
                            }
                            break // break out of the for loop once delete the item
                        }
                    }
                }
            }
        }catch{
            print("error")
        }
    }
}
}
```