

```

//
// Reserve.swift
// Smithington Public High School Library
//
// Created by Colten SeEVERS & Nick Kortz on 1/29/18.
// Copyright © 2018 Colten & Nick Kortz. All rights reserved.
//
//

import UIKit

class Reserve: UIViewController, UITableViewDataSource, UITableViewDelegate,
UISearchBarDelegate {

    @IBOutlet var table: UITableView!
    @IBOutlet var searchBar: UISearchBar!

    @IBOutlet weak var CurrentUserLabel: UILabel!

    var currentBookArray = [Book]() //update table
    var CurrentSelectedBook = ""
    var CurrentBook = Book()
    var index: Int = 0

    override func viewDidLoad() {
        super.viewDidLoad()
        currentBookArray = bookArray
        CurrentUserLabel.text = CurrentUser
        setUpSearchBar()
        table.estimatedSectionHeaderHeight = 90
        searchBar.placeholder = "Search Book by Name"
    }

    @IBAction func CheckoutButton(_ sender: UIButton) {
        //Set up Checkout Period
        let futureDate = Calendar.current.date(byAdding: dateComponent, to:
            CurrentDate)
        let calendar = Calendar.current
        let year = calendar.component(.year, from: futureDate!)
        let month = calendar.component(.month, from: futureDate!)
        let day = calendar.component(.day, from: futureDate!)

        if CurrentBook.barcode != "" && CurrentBook.status != .out
        {
            CurrentDue = (String(month)+":"+String(day)+":"+String(year))

            //Create the Alert
            let alert = UIAlertController(title: "Book Checkout", message:
                CurrentBook.name + " has been checked out. Due Date:" + CurrentDue,
                preferredStyle: UIAlertControllerStyle.alert)
            //Add an Action (button)
            alert.addAction(UIAlertAction(title: "OK", style:
                UIAlertActionStyle.default, handler: nil))
            self.present(alert, animated: true, completion: nil)
        }
    }
}

```

```

index = 0
for book in bookArray{
    if(book.barcode == CurrentBook.barcode)
    {
        let oldname = book.name
        let oldcategory = book.category
        let oldbarcode = book.barcode
        bookArray.remove(at: index)
        bookArray.append(Book(name: oldname, category: oldcategory,
            barcode: oldbarcode, status: .out, duedate: CurrentDue))
        CurrentBookList = CurrentBook
        //Update Database with Checked Out Book
        TableCheckout(CurrentCode: CurrentCode, CObook:
            oldname, CObarcode: oldbarcode, COdue: CurrentDue)
        CObook = oldname
        CObarcode = oldbarcode
        COdue = String(CurrentDue)
    }
    index+=1
}
}
else if CurrentBook.barcode != "" && CurrentBook.status == .out
{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message: "The
        Selected Book is Already Checked Out, Please Select another Book",
        preferredStyle: UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
else
{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message: "No Book
        has been Selected, Please Select a Book", preferredStyle:
        UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
}

@IBAction func ReserveButton(_ sender: UIButton) {
    _ = UITableViewRowAction(style: .default, title: "Reserve") { (action,
        indexPath) in }
    if CurrentBook.barcode != "" && CurrentBook.status == .available
    {
        let alert = UIAlertController(title: "Book Reserved", message:
            CurrentBook.name + " has been Reserved.", preferredStyle:
            UIAlertControllerStyle.alert)
        alert.addAction(UIAlertAction(title: "OK", style:
            UIAlertActionStyle.default, handler: nil))
    }
}

```

```

        self.present(alert, animated: true, completion: nil)
        index = 0
        for book in bookArray{
            if(book.barcode == CurrentBook.barcode)
            {
                let oldname = book.name
                let oldcategory = book.category
                let oldbarcode = book.barcode
                bookArray.remove(at: index)
                bookArray.append(Book(name: oldname, category: oldcategory,
                    barcode: oldbarcode, status: .reserve, duedate: ""))
                CurrentReservedList = CurrentBook
                TableReserve(CurrentCode: CurrentCode, RESbook: oldname,
                    RESbarcode: oldbarcode)
                GrabInfo()

                RESbook = ""
                RESbarcode = ""
            }
            index+=1
        }

    }
    else{
        //Create the Alert
        let alert = UIAlertController(title: "Nothing has been Selected",
            message:"No Book has been Selected, Please Select a Book",
            preferredStyle: UIAlertControllerStyle.alert)
        //Add an Action (button)
        alert.addAction(UIAlertAction(title: "OK", style:
            UIAlertActionStyle.default, handler: nil))
        self.present(alert, animated: true, completion: nil)
    }

}

private func setUpSearchBar() {
    searchBar.delegate = self
}

// Table

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

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    guard let cell = tableView.dequeueReusableCell(withIdentifier: "Cell") as?
        TableCell else {
        return UITableViewCell()
    }

    cell.nameLbl.text = currentBookArray[indexPath.row].name

```

```

        cell.categoryLbl.text = currentBookArray[indexPath.row].category.rawValue
        cell.barcodeLbl.text = currentBookArray[indexPath.row].barcode
        cell.statusLbl.text = currentBookArray[indexPath.row].status.rawValue
        return cell
    }

```

```

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) ->
    CGFloat {
    return 85
}

```

```

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    if (currentBookArray[indexPath.row].barcode != ""){
        CurrentBook = currentBookArray[indexPath.row]
    }
}

```

```

func tableView(_ tableView: UITableView, editActionsForRowAt indexPath:
    IndexPath) -> [UITableViewRowAction]? {

```

```

    let Checkout = UITableViewRowAction(style: .default, title: "Checkout")
    { (action, indexPath) in
        //Set up Checkout Period
        let futureDate = Calendar.current.date(byAdding: dateComponent, to:
            CurrentDate)
        let calendar = Calendar.current
        let year = calendar.component(.year, from: futureDate!)
        let month = calendar.component(.month, from: futureDate!)
        let day = calendar.component(.day, from: futureDate!)

        //Select the book from the Swipe
        self.CurrentBook = self.currentBookArray[indexPath.row]

        if self.CurrentBook.barcode != "" && self.CurrentBook.status != .out
        {
            CurrentDue = (String(month)+":"+String(day)+":"+String(year))

            //Create the Alert
            let alert = UIAlertController(title: "Book Checkout", message:
                self.CurrentBook.name + " has been checked out. Due Date:" +
                CurrentDue, preferredStyle: UIAlertControllerStyle.alert)
            //Add an Action (button)
            alert.addAction(UIAlertAction(title: "OK", style:
                UIAlertActionStyle.default, handler: nil))
            self.present(alert, animated: true, completion: nil)
            self.index = 0
            for book in bookArray
            {
                if(book.barcode == self.CurrentBook.barcode)
                {
                    let oldname = book.name
                    let oldcategory = book.category
                    let oldbarcode = book.barcode

```

```

        bookArray.remove(at: self.index)

        bookArray.append(Book(name: oldname, category: oldcategory,
            barcode: oldbarcode, status: .out, duedate: CurrentDue))
        CurrentBookList = self.CurrentBook
        CObook = oldname
        CObarcode = oldbarcode
        COdue = String(CurrentDue)
        TableCheckout(CurrentCode: CurrentCode, CObook:
            oldname, CObarcode: oldbarcode, COdue: CurrentDue)
    }
    self.index+=1
}
}
else if self.CurrentBook.barcode != "" && self.CurrentBook.status
    == .out
{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message: "The
        Selected Book is Already Checked Out, Please Select another Book",
        preferredStyle: UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
    self.table.reloadData()
}
else
{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message: "No
        Book has been Selected, Please Select a Book", preferredStyle:
        UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
}
let Checkin = UITableViewRowAction(style: .default, title: "Return")
{ (action, indexPath) in

    //Select the book from the Swipe
    self.CurrentBook = self.currentBookArray[indexPath.row]

    if self.CurrentBook.barcode != "" && self.CurrentBook.status == .out
    {

        //Create the Alert
        let alert = UIAlertController(title: "Book Returned", message:
            self.CurrentBook.name , preferredStyle:
            UIAlertControllerStyle.alert)
        //Add an Action (button)
        alert.addAction(UIAlertAction(title: "OK", style:
            UIAlertActionStyle.default, handler: nil))
    }
}

```

```

self.present(alert, animated: true, completion: nil)
self.index = 0
for book in bookArray
{
    if(book.barcode == self.CurrentBook.barcode)
    {
        let oldname = book.name
        let oldcategory = book.category
        let oldbarcode = book.barcode
        bookArray.remove(at: self.index)
        bookArray.append(Book(name: oldname, category: oldcategory,
            barcode: oldbarcode, status: .available, duedate: ""))
        // CurrentBookList = self.CurrentBook
        TableCheckIN(CurrentCode: CurrentCode)
        CObook = ""
        CObarcode = ""
        COdue = ""
    }
    self.index+=1
}
}
else {
    //Create the Alert
    let alert = UIAlertController(title: "Book is not Checkout",
        message: "The Selected Book is not Checked Out", preferredStyle:
        UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
    self.table.reloadData()
}
}
let Reserve = UITableViewRowAction(style: .default, title: "Reserve")
{ (action, indexPath) in
    self.table.reloadData()

    //Select the book from the Swipe
    self.CurrentBook = self.currentBookArray[indexPath.row]
    if self.CurrentBook.barcode != "" && self.CurrentBook.status
        == .available
    {
        let alert = UIAlertController(title: "Book Reserved", message:
            self.CurrentBook.name + " has been Reserved.", preferredStyle:
            UIAlertControllerStyle.alert)
        alert.addAction(UIAlertAction(title: "OK", style:
            UIAlertActionStyle.default, handler: nil))
        self.present(alert, animated: true, completion: nil)
        self.index = 0
        for book in bookArray{
            if(book.barcode == self.CurrentBook.barcode)
            {
                let oldname = book.name
                let oldcategory = book.category

```

```

        let oldbarcode = book.barcode
        bookArray.remove(at: self.index)
        bookArray.append(Book(name: oldname, category: oldcategory,
            barcode: oldbarcode, status: .reserve, duedate: ""))
        CurrentReservedList = self.CurrentBook
        TableReserve(CurrentCode: CurrentCode, RESbook: oldname,
            RESbarcode: oldbarcode)
        RESbook = oldname
        RESbarcode = oldbarcode
    }
    self.index+=1
}
self.table.reloadData()

}
else{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message:"Book
        is Currently Checkout or Reserved", preferredStyle:
        UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
        UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
}
let Unreserve = UITableViewRowAction(style: .default, title: "Unreserve")
{ (action, indexPath) in
    self.table.reloadData()

    //Select the book from the Swipe
    self.CurrentBook = self.currentBookArray[indexPath.row]
    if self.CurrentBook.barcode != "" && self.CurrentBook.status == .reserve
    {
        let alert = UIAlertController(title: "Book Unreserved", message:
            self.CurrentBook.name + " has been unreserved.", preferredStyle:
            UIAlertControllerStyle.alert)
        alert.addAction(UIAlertAction(title: "OK", style:
            UIAlertActionStyle.default, handler: nil))
        self.present(alert, animated: true, completion: nil)
        self.index = 0
        for book in bookArray{
            if(book.barcode == self.CurrentBook.barcode)
            {
                let oldname = book.name
                let oldcategory = book.category
                let oldbarcode = book.barcode
                bookArray.remove(at: self.index)
                bookArray.append(Book(name: oldname, category: oldcategory,
                    barcode: oldbarcode, status: .available, duedate: ""))
                CurrentReservedList = self.CurrentBook
                TableUNReserve(CurrentCode: CurrentCode)
                self.CurrentBook = Book()
                RESbook = ""
            }
        }
    }
}

```

```

        RESbarcode = ""
    }
    self.index+=1
}
self.table.reloadData()

}
else{
    //Create the Alert
    let alert = UIAlertController(title: "Book Checkout", message:"Book
    is Currently Checkout or Reserved", preferredStyle:
    UIAlertControllerStyle.alert)
    //Add an Action (button)
    alert.addAction(UIAlertAction(title: "OK", style:
    UIAlertActionStyle.default, handler: nil))
    self.present(alert, animated: true, completion: nil)
}
}

Checkout.backgroundColor = UIColor.red
Reserve.backgroundColor = UIColor.blue
Unreserve.backgroundColor = UIColor.orange
Checkin.backgroundColor = UIColor.purple

print(self.CurrentBook.status)
return[Reserve, Checkout,Checkin,Unreserve]
}

```

// Search Bar

```

func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
    currentBookArray = bookArray.filter({ Book -> Bool in
        switch searchBar.selectedScopeButtonIndex {
        case 0:
            if searchText.isEmpty { return true }
            return Book.name.lowercased().contains(searchText.lowercased())
        case 1:
            if searchText.isEmpty { return Book.category == .fiction }
            return Book.name.lowercased().contains(searchText.lowercased()) &&
                Book.category == .fiction
        case 2:
            if searchText.isEmpty { return Book.category == .nonfiction }
            return Book.name.lowercased().contains(searchText.lowercased()) &&
                Book.category == .nonfiction
        case 3:
            if searchText.isEmpty { return Book.category == .nonfiction }
            return Book.name.lowercased().contains(searchText.lowercased()) &&
                Book.category == .historical
        case 4:
            if searchText.isEmpty { return true }
            return Book.barcode.lowercased().contains(searchText.lowercased())
        case 5:
            if searchText.isEmpty { return Book.category == .fiction }

```



```

        return Book.barcode.lowercased().contains(searchText.lowercased()) &&
            Book.category == .fiction
    case 6:
        if searchText.isEmpty { return Book.category == .nonfiction }
        return Book.barcode.lowercased().contains(searchText.lowercased()) &&
            Book.category == .nonfiction
    case 7:
        if searchText.isEmpty { return Book.category == .nonfiction }
        return Book.barcode.lowercased().contains(searchText.lowercased()) &&
            Book.category == .historical
    default:
        return false
    }
})
table.reloadData()
}

```

```

func searchBar(_ searchBar: UISearchBar, selectedScopeButtonIndexDidChange
selectedScope: Int) {
    switch selectedScope {
    case 0:
        currentBookArray = bookArray
    case 1:
        currentBookArray = bookArray.filter({ Book -> Bool in
            Book.category == .fiction
        })
    case 2:
        currentBookArray = bookArray.filter({ Book -> Bool in
            Book.category == .nonfiction
        })
    case 3:
        currentBookArray = bookArray.filter({ Book -> Bool in
            Book.category == .historical
        })
    default:
        break
    }
    table.reloadData()
}
}

```