

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

## Assignment 4

### Testing Environment

- Your app will be tested using the **iPhone 11**. Before submitting, please ensure your application looks correct on this device.

### Submission Checklist

#### Before you submit:

1. In your Firebase project, add the provided email address as an **Editor**. You can find this option under Project Overview > Project Settings > Users & Permissions. Submissions that do not include this email as an Editor will automatically be graded 0.
2. Zip your entire project and name the file **a4-studentname-username.zip**
  - a. Replace **studentname** and **username** with your name and id
  - b. No .7zip or .rar files accepted
  - c. Ensure that your project contains the **GoogleInfo.plist** associated with your Firebase project.
3. Prepare a screen recording demonstrating the functionality you implemented. In your screen recording, you should provide a verbal description of what you are doing on the screen (as you do it). In your screen recording, you should show your **app running side by side with your Firestore console**. You should demo how your application is affecting the Firestore database.

#### In the assignment dropbox:

1. Upload your screen recording to **OneDrive** and ensure that the link is shared with the instructor.
2. Submit your zip file containing the project
3. In the submission comments, provide:
  - a. The email address associated with your Firebase project. This will be used to locate your database.
  - b. The OneDrive link containing your screen recording.

For your submission to be graded, you must provide all the above information.

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

### Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
  - Communication with others (both inside and outside the class)
  - Discussion of solution or approaches with others; sharing/using a “reference” from someone
  - Searching the internet for full or partial solutions
  - Sharing of resources, including links, computers, accounts

### Grading Criteria

- The majority of grades are assigned based on the correct completion of the required functionality.
- Submissions are required to use the correct coding conventions used in class, professional organization of the code, alignment, clarity of names will all be part of the evaluation.
- The user interface of your application must be reasonably polished, easy to understand, and readable. Use reasonably pleasant colors and typography.

## Problem Description

Build an iOS application for the local library. Users can view a list of books, borrow a book, and return a book.

### Screen 1: Login Screen

1. When the app loads, display a login screen. The login screen must contain textboxes for the user to enter their library username and password, and a label for displaying error messages.
2. For simplicity, assume there are only two stored users:
  - username = psmith, password = 1234
  - username = tbrown, password = 0000
2. Usernames and passwords must be stored in a **Firestore** collection. (NOTE: You are not permitted to use any Firebase services related to login/users, such as FirebaseAuth)
3. If the user enters a valid username/password combination, navigate them to Screen #2. This must be performed via a Navigation Controller. You must also **pass the username** to Screen #2.
4. If the user enters an **invalid** username/password, display an error message in a **label** and do not let them proceed.

### Screen 2: List of Library Books

1. Screen 2 displays a list of books in a tableview and a label for showing error messages. The tableview must show the list of books in the library. ~~Tapping on the logout button must "logout" the user and return them back to Screen #1~~
2. If the book is available for borrowing, display the book's title and author. If the book is *not* available, then display the book title, author, and the username of the person who borrowed the book.
  - The layout of each row should be the built-in **Subtitle** layout.
  - **The list of books must be read from a Firestore collection.**

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

2. Every book must be represented as a **struct** and must contain the following properties:
  - book title
  - book author
  - the username of the person who borrowed the book. **Not every book will be borrowed!**
  - availability (checked out vs. available). **This must be represented as a computed property** → if the book is associated with a user, then the book is borrowed.
3. The user can borrow a book by tapping on a row in the book list.
  - The book can be borrowed if it is available (no other user has borrowed the book)
  - If a book can be borrowed, update the book's status in Firestore and in the tableview.
  - If the book cannot be borrowed, display an error message in a **label**.
4. The user can return their book by swiping left on a row in the tableview and pressing the **Delete** button. If the book can be returned, update the book's status in Firestore and tableview.
5. The book can only be returned if:
  - The selected book was borrowed by the currently logged in user.
  - If the book cannot be returned, then display an error message
5. Tapping the *Back* button will return the user to Screen #1

**END OF ASSESSMENT**