

Mobile App Development

Assignment 10

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a) Assignment #.
 - b) File Name.
 - c) Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. Submit Codes:
 - a) Zip all the project folder to be submitted on canvas.
4. Submission details:
 - a) The file name is very important and should follow the following format:
Assignment#.zip
 - b) You should submit the assignment through Canvas: Submit the zip file.
5. **Failure to follow the above instructions will result in point deductions.**

Assignment 10 (100 Points)

In this assignment you will implement a grade tracking application. This app uses Firebase for Authentication and Data storage.

1. Create a new Firebase project.
2. Import Firebase into the provided skeleton app make sure to register the app in your Firebase project.
3. Import Firebase Authentication and Firestore libraries into your project.
4. All the flow and UI parts of the app have been included in the skeleton project, you should focus on making the required Firebase Auth and Firestore calls to implement the required functions.

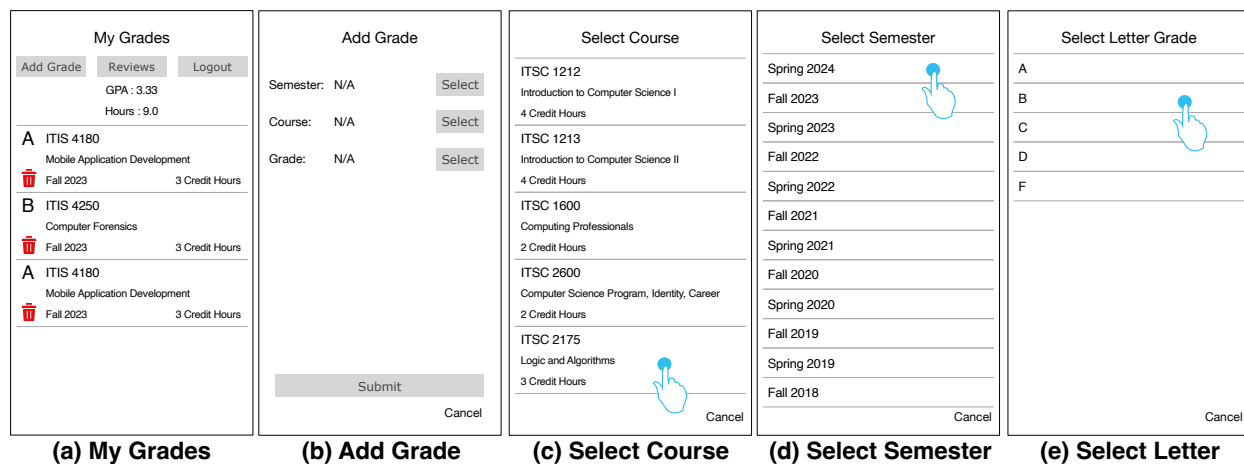


Figure 1, Application User Interface

Part 1: Authentication (0 Points)

The requirements are as follows:

1. This feature is already provided in the skeleton app provided.

Part 2 : My Grades Screen (20 Points)

The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

1. Retrieve the list of grades for the currently logged in user from Firestore.
 - a. You should create a Grade class to store the grade information.
 - b. Setup a snapshot listener to listen to realtime updates for the grades collection for only the logged in user and to update the displayed grades list when the grades are updated.
2. The overall GPA and number of course hours should be displayed as shown Fig 1(a).
 - a. The next section describes how to compute the GPA.
 - b. If the number total number of hours taken is 0 then set the GPA to 4.0
3. Upon clicking the delete icon the following operations should be performed:
 - a. The selected course should be deleted from the firebase.
 - b. The courses list should be reloaded from firebase, and the GPA and the Hours should be updated. Note, this is handled through the snapshot listener.
4. Clicking the "Add Grade" button should show the "Add Grade" Screen.
5. Clicking the "Reviews" button should show the "Course Reviews" Screen.

Part 3 : Add Grade Screen (20 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. All the transitions and select buttons have been implemented in the provided skeleton app.
2. If the user enters all the course fields and clicks submit, the app should save the newly created grade to firebase firestore and associate it with the currently logged in user. Then go back to the Grades Screen, which should refresh the list of grades and update the GPA and hours listed.

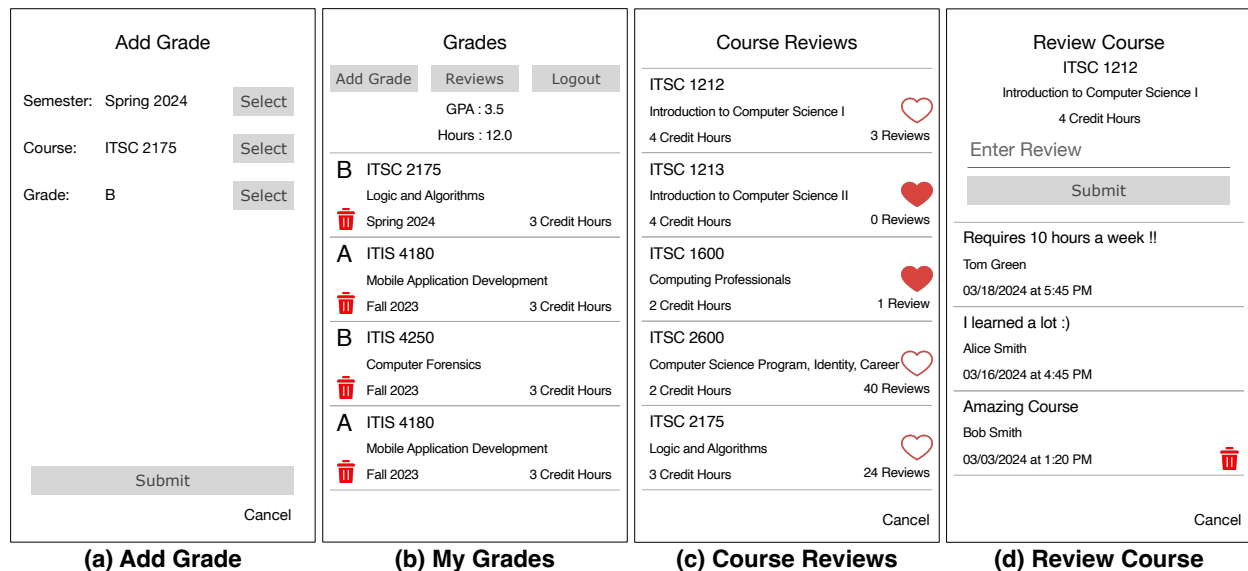


Figure 2, Application User Interface

Part 4 : Course Reviews Screen (40 Points)

The interface should be created to match the UI presented in Figure 2(c). The requirements are as follows:

1. The list of courses should be retrieved by making a call to the get courses API provided. URL: <https://www.theappsdr.com/api/cji-courses> and Method: GET
2. Upon loading this screen make an API call using your library of choice to retrieve the list of courses.
 - a. When the API request is done, parse the course JSON and stored the parsed objects in an array of course objects.
 - b. The displayed course items list is shown in Figure 2(c).
 - c. The filled heart and not-filled heart icons to indicate if this course has been favorited or no by the current logged-in user.
3. Use Firebase Firestore to manage the course review and favorited by information:
 - a. Create a **SnapshotListener** to retrieve the course collection which should host course information, reviews and the uid of the users who have favorited a given course.
 - b. The retrieved courses should be stored in a separate array or set, which should

- be used by the displayed list adapter to check if whether a displayed course item has been favorited by the currently logged in user or no. Where courses in the user's favorites should show filled heart icon, and otherwise should show not-filled heart icon.
- c. Note the number of reviews presented, which should be stored as an integer in the course document on Firestore. The number should be updated once a review is created or is deleted.
- 4. If the user clicks on the not-filled heart icon (add to favorites):
 - a. Add/Update the current user uid to the course information on Firebase to indicate that course is favorited by the user.
 - b. Upon Firestore's successful update, the SnapshotListener will capture the change and the displayed list should be refreshed to reflect this change, where the not-filled heart icon should change to a filled heart icon to indicate the course has been favorited by the currently logged in user.
- 5. If the user clicks on the filled heart icon (delete from favorites):
 - a. Delete the current user uid from the course information on Firebase to indicate that course is not favorited by the user.
 - b. Upon Firestore's successful update, the SnapshotListener will capture the change and the displayed list should be refreshed to reflect this change, where the filled heart icon should change to a not-filled heart icon to indicate the course has been not favored by the currently logged in user.
- 6. Clicking on a row item should go to the Review Course Screen.

Part 5 : Review Course Screen (20 Points)

The interface should be created to match the UI presented in Figure 2(d). The requirements are as follows:

1. At the top of the screen, the course information.
2. Retrieve the reviews for the course from Firestore.
 - a. Setup a snapshot listener to listen to realtime updates for the reviews sub-collection which is under the course document.
 - b. The displayed list should be updated when the snapshot listener is triggered. **The trash "Delete" icon should only be displayed for review items that were created by the currently logged in user.**
3. If the user clicks the "Delete" icon the review should be deleted from the course's reviews sub-collection from Firestore. In addition, decrement the reviews count in the course collection. The previously setup realtime updates which should be triggered which should refresh the reviews list to reflect this change.
4. Creating a Review:
 - a. The EditText should allow the user to enter a text review. Clicking the "Submit" button should check if there is a review entered.
 - b. If there is a review entered then save the review on Firestore under the courses review sub-collection as a new review document. The previously setup realtime updates which should be triggered which should refresh the comments list to reflect this change.
 - c. Increment the review count in the course document.

How to compute the GPA?

The grade point average (GPA) is calculated by dividing the total amount of grade points earned by the total amount of credit hours attempted. Your grade point average may range from 0.0 to a 4.0. Each letter grade is assigned a grade point as shown below:

A = 4.0 grade points

B = 3.0 grade points

C = 2.0 grade points

D = 1.0 grade points

F = 0.0 grade points

Example Student Grades			
Course	Credit Hours	Grade	Grade Points
Biology	3	A	12
Biology Lab	1	B	3
English 101	3	C	6
Maths 101	3	A	12
10 Total Credit Hours Completed		33 Total Grade Points	
GPA = 33/10 = 3.3			

$$GPA = \frac{\text{Total Grade Points}}{\text{Total Credit Hours}}$$