Mobile App Development
In-Class Assessment 4

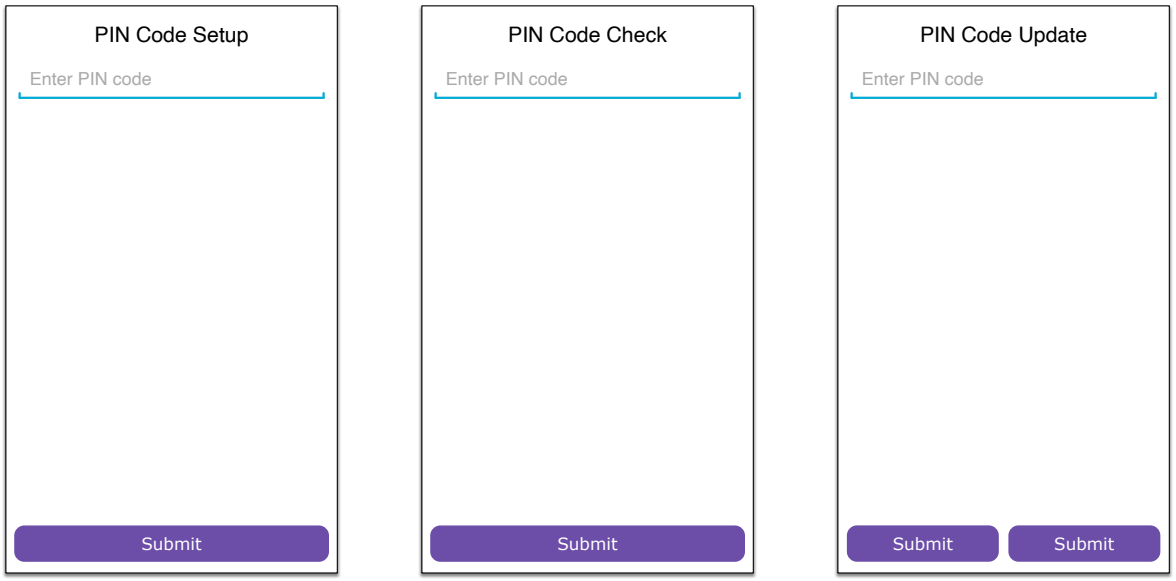
Basic Instructions:

1. This is and In Class Assessment, which will count for 8% of the total course grade.
2. This assessment is an individual effort. Each student is responsible for her/his own assessment and its submission.
3. Once you have picked up the assessment, you may not discuss it in any way with anyone until the assessment period is over.
4. During the assessment, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other online persons.
5. Answer all the assessment parts, all the parts are required.
6. During the assessment the teaching assistants and Instructors will pass by each student and ask them to demonstrate their application. Your interaction with the teaching assistants and instructors will be taken into consideration when grading your assessment submission.
7. Please download the support files provided with the assessment and use them when implementing your project.
8. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
9. Create a zip file which includes all the project folder, any required libraries, and your presentation material. Submit the exported file using the provided canvas submission link.
10. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

In-Class Assessment 4 (100 Points)

In this assignment you will build as simple grades tracking app. You are provided with the skeleton app that includes the basic app flow and layouts. This app uses Room library to store and manage the user grades.

1. The pin code should be stored locally using the **Shared Preferences**. https://developer.android.com/training/data-storage/shared-preferences
2. The grades should be stored and retrieved using the Room Library.
3. All communication between fragments should be performed using interfaces.



(a) Pin Code Setup Fragment (b) Pin Code Check Fragment (c) Pin Code Update Fragment

Figure 1, App Wireframe

Part 1: Main Activity (5 Points)

The requirements are as follows:

1. *Upon starting the Main Activity should check if the user has setup a pin code or no, This should be done by checking if a pin code is currently stored in the Shared Preferences or not.*
2. If no pin code is stored in the Shared Preferences then display the Pin Code Setup Fragment.
3. If a pin code is stored in the Shared Preferences then display the Pin Code Check Fragment.

Part 2: Pin Code Setup Fragment (5 Points)

This fragment is displayed if no pin code is currently stored in the Shared Preferences. The requirements are as follows:

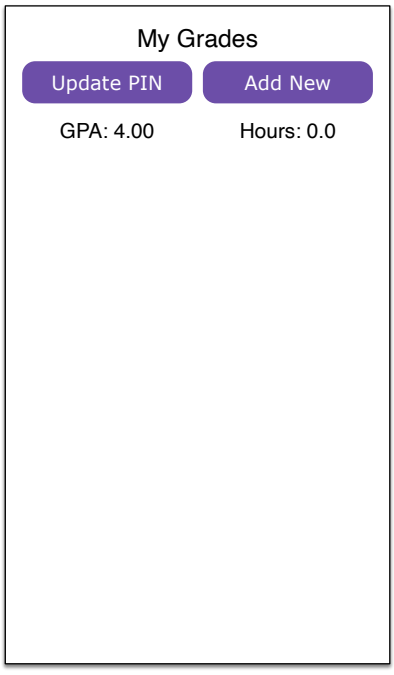
1. *Upon clicking the submit button, the selected pin code should be sent to the Main Activity using "onPinCodeSetup" method, which should:*
   a. Store the pin code in the Shared Preferences.
   b. Replace this fragment with the Pin Code Check Fragment.

Part 3: Pin Code Check Fragment (5 Points)
This fragment is displayed if a pin code is currently stored in the Shared Preference. The requirements are as follows:

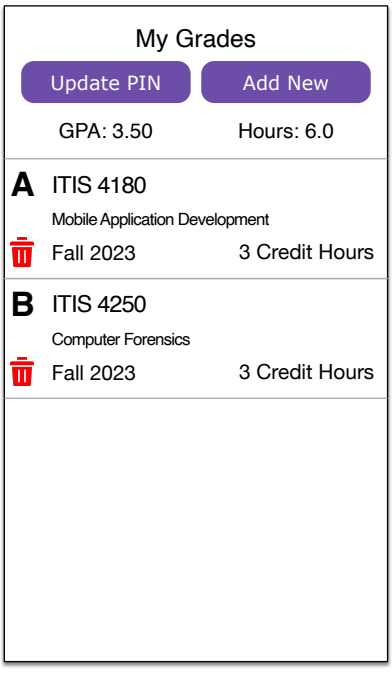
1. This screen should allow the user to enter a pin code, which should be compared with the previously stored pin code in the Shared Preferences.
2. *Upon clicking the submit button, the entered pin code should be sent to the Main Activity, which should:*
   a. *Implement the "checkPinCode" method, which should read the pin code stored in the Shared Preferences, and compare it to the entered pin.*
      i. If the pin code match then return true.
      ii. If the pin codes do not match then return false.
   b. We implemented the the logic required to load the My Grades fragment when the pin codes match.

Part 4: Pin Code Update Fragment (5 Points)
This fragment allows the user to update the pin code. The requirements are as follows:

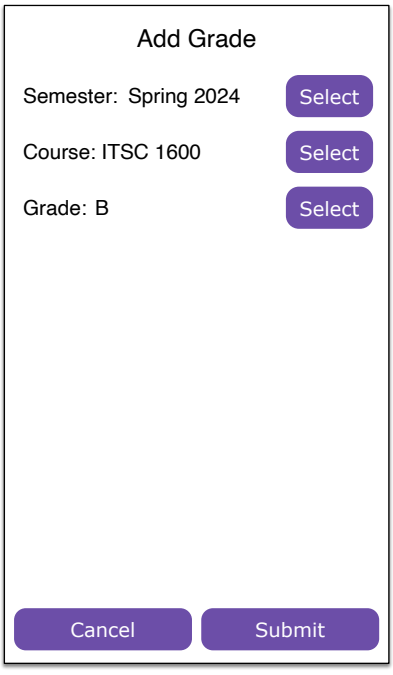
1. *Upon clicking the submit button, the selected pin code should be sent to the Main Activity using "onPinCodeUpdate", which should:*
   a. Store the pin code in the Shared Preferences.
   b. Replace this fragment with the Pin Code Check Fragment.
2. Upon clicking the cancel button communicated with the Main Activity, which should:
   a. Replace this fragment with the Grades Fragment.



(a) My Grades Fragment



(b) My Grades Fragment



(c) Add Grade Fragment

Figure 2, App Wireframe

Setup the Entities and Data Access Objects required to manage the Grade storage information using the Rooms Database.

Part 5 : Add Grade Fragment (30 Points)

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

1. All the layout and logic for this fragment has been created in the provided support app.
2. Clicking on the Submit button should:
   a. Create a new Grade object using the selected course, semester and letter grade.
   b. In the Main Activity implement the "onAddGrade" method which should:
      i. Store the new Grade object in the Rooms Database.
      ii. Pop the back stack to go back to the My Grades Fragment.

Part 6 : My Grades Fragment (50 Points)

Fragment displays the grades that are stored in Rooms database. This fragment is shown in Figure 2(a)(b). The requirements are as follows:

1. Clicking the "Update Pin" button should:
   a. Replace this fragment by the Pin Code Update Fragment.
2. *Implement the Main Activity method "**getAllGrades**" which should:*
   a. Retrieve all the list of grade objects from the Rooms Database and return an Array List of grades.
3. *Create a RecyclerView Adapter that displays the retrieved Grades:*
   a. Each row item displays the letter grade, course number, course name, semester, and number of credit hours.
4. *Calculate and display the GPA and the total hours as shown in Figure 2(b).*
   a. Implement the method "***calculateAndDisplayGPA***" which should calculate and display the GPA and total hours.
5. *Clicking the trash can in a row item should delete the selected grade by using the interface and calling "**deleteGrade**" method which should:*
   a. Delete the selected Grade from the Rooms Database.
   b. Refresh the RecyclerView to display the updated grade lists.
   c. Refresh the displayed GPA and total hours.

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			

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

Student Name:	
Student ID:	

Part #	Features	Total	Grade
Part 1	Main Activity checks if a pin is setup or no in the Shared Preferences and load the correct fragment.	5	
Part 2	Pin Code Setup implemented correctly and pin is stored locally. Error handling implemented correctly.	5	
Part 3	Pin Code Check implemented correctly, entered pin is compared to locally stored pin. Error handling implemented correctly.	5	
Part 4	Pin Code Update implemented correctly and pin is updated locally. Error handling implemented correctly.	5	
Part 5	Setup the Entity, DAO and Database required to use the Rooms Library to store the Grade information.	10	
Part 5	**Add Grade Fragment:** implemented the ***onAddGrade*** method to store the Grade in the Room Database, then pop the back stack.	20	
Part 6	**My Grades Fragment:** • Implemented the ***getAllGrades*** method to retrieve the list of grades stored in the rooms database. • Implement the RecyclerView Adapter to display the grades retrieved. • Implement the ***calculateAndDisplayGPA*** to display the GPA and total hours.	40	
Part 6	**My Grades Fragment:** • Clicking on the trash can icon should delete the Grade and refresh the RecyclerView. • Implemented the ***deleteGrade*** method.	10	
	Total	**100**	

Table 1: Grading Key