Mobile App Development

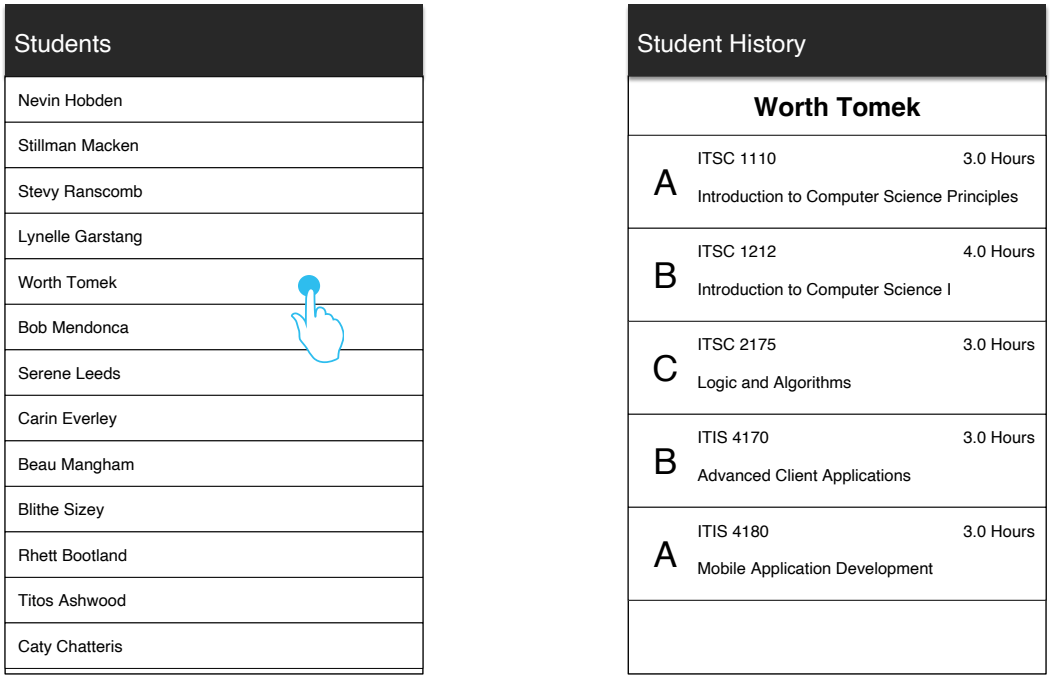
In-Class Assessment 4 (2 Hours)

Basic Instructions:

1. This is an In Class Assessment, which counts for 10% 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)

You are provided with a skeleton application which includes all the layout files and classes required for this assignment, so please use the provided skeleton app.



(a) Students Fragment **(b) Student History Fragment**

Figure 1, Application User Interface

Part 1, Students Fragment (40 Points):

This fragment is shown in Fig 1(a). The requirements are listed below:

1. This fragment displays the list of Students in a ListView as shown in Fig 1(a).
   a. Use the *DataServices.getStudents()* method to get the list of students from the DataServices class.
   b. You can use the default ArrayAdapter to setup the adapter for the ListView as shown in Fig 1(a), where each row item should display the student name.
2. Clicking on a student row item should should communicate with the Main Activity using an interface in order to:
   a. Send the selected Student object to the Main Activity.
   b. Replace the current fragment with the Student History fragment.
   c. Push the current fragment on the back stack.

Part 2, Student History Fragment (60 Points):

This fragment is shown in Fig 1(b). The requirements are listed below:

1. This fragment receives a Student object, which contains the student name, and a list of Course History objects representing the courses taken by the students and the grades earned in each course *(call student.getCoursesHistory()).*
2. This fragment displays the list of Student's name at the top of the fragment as shown in Fig 1(b).
3. Create a custom Array Adapter to display the Students course history, where each row item should show the course name, course number, course hours, and the grade earned in this course as shown in Fig 1(b).

Section:	
Student Name:	
Student ID:	

Part #	Features	Total	Grade	Comments
P1	Used default ArrayAdapter and displayed the students in a ListView as showing each student's name.	30		
P1	Clicking Student row item: - Sends the selected Student to the Student History fragment - Shows the Student History fragment - Pushes the current fragment back stack. **Interface Implementation REQUIRED**	10		
P2	Display the student's name at the top of the screen as shown in Fig 1(b).	10		
P2	Created a custom ArrayAdapter to display the selected student's course history. ListView showing the student history as shown in Fig 1(b).	50		
	Total	**100**		

Table 1: Grading Key