CEN4535C Development of Gaming and Mobile Applications
Fall 2016
Android Programming Exam
Ching-Hua Chuan (c.chuan@unf.edu)
Due: September 28 (Wednesday, end of day)
Total points: 40

Instructions

This is a take-home exam. You must write the Android program <u>by yourself</u> for the following question. To submit your program, please compress the files in the project into a zip file and upload the zip file via <u>Canvas</u>. No email submissions.
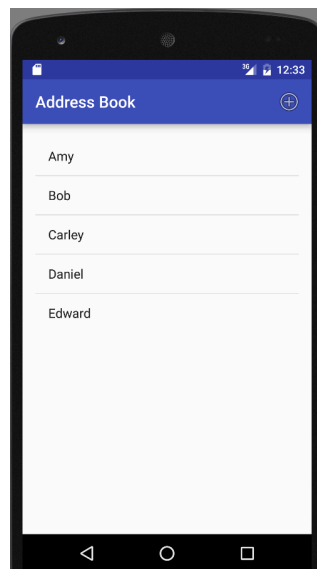
Your program will be checked for software plagiarism. <u>If your program was found to be very similar to another student's, the grade will be shared by the two students.</u> For example, if two submissions were detected as very similar and one of the submissions scores 40 points, the two students responsible for the two submissions will receive 20 points each. By the same token, if three programs are found to be highly similar, each person will receive 13.33 points.
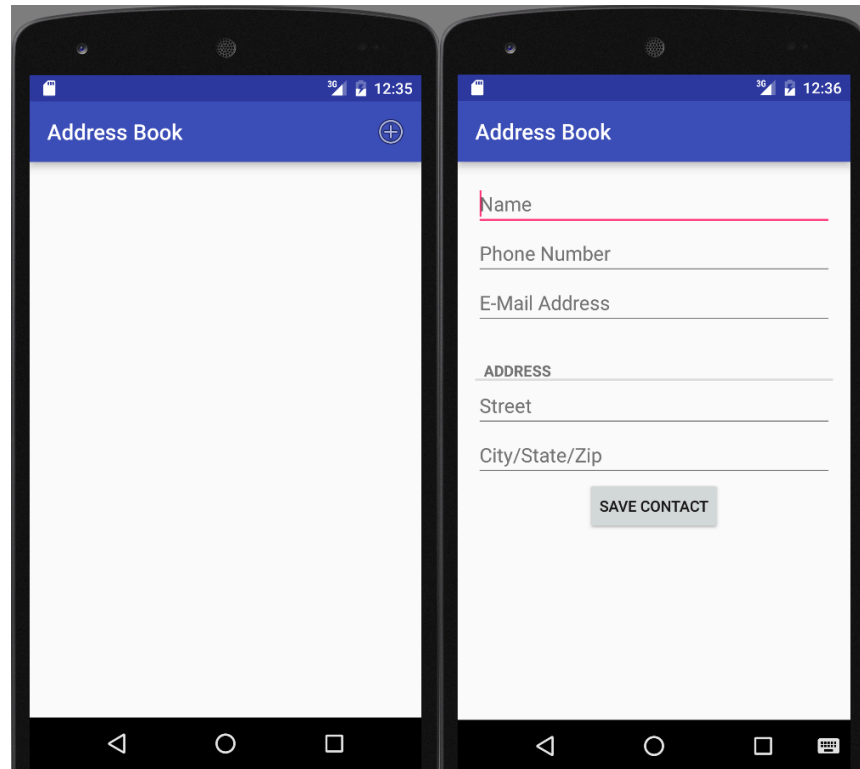
Late submissions

No late submissions will be accepted.

Address Book App

Create an address book app that allows users to add, edit, and delete contacts. The first view of the app should look like the following screenshot:

When the user first runs the app, there may be no contacts on the list. To add a new contact, the user would click the menu button that will bring the Add Contact option as follows:
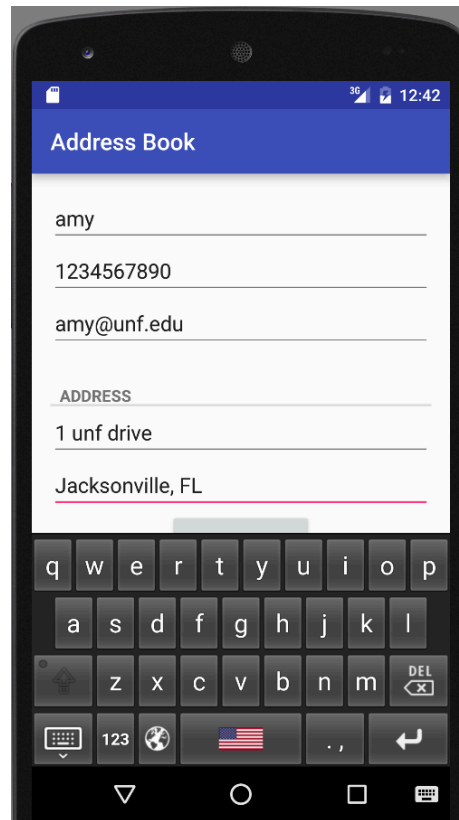

Note that "Add Contact" option will be shown in action bar for devices running Android API Level 11 or higher.

Once the user touches the Add Contact option, the app should bring a new view that allows the user to add detailed contact information. Make sure the Add Contact view is scrollable so that the keyboard is not blocking the space.

When the user finishes adding contact information and clicks the Save Contact button, the app should bring back the first view and show the newly added contact on the list. If the user touches any contact on the list, the app brings up context menus with two options: Edit and Delete.
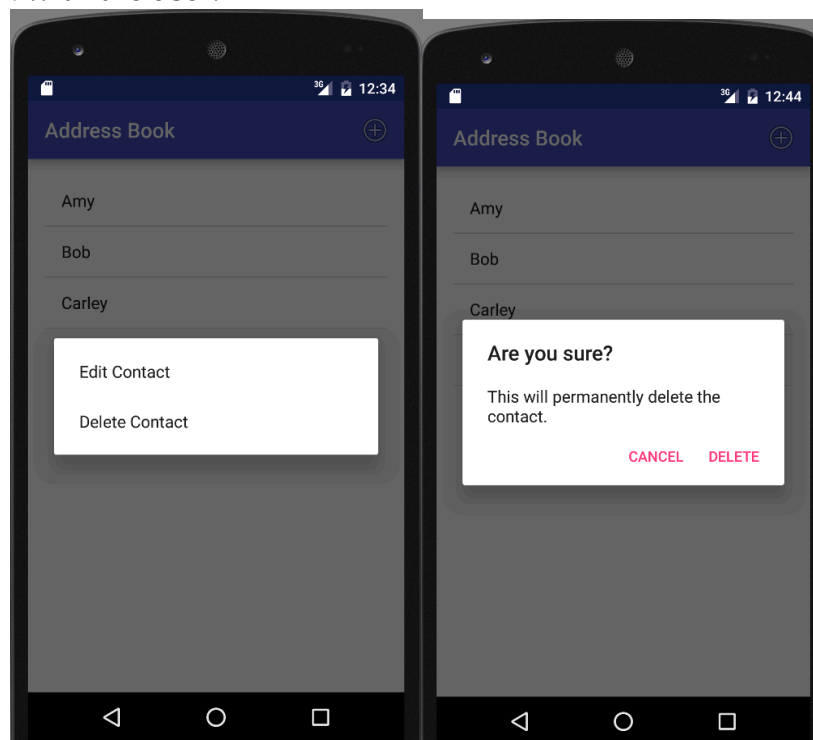
Edit Contact
The Edit Contact option allows the user to edit an existing contact. It can use the same view for adding a new contact with pre-populated contact information.
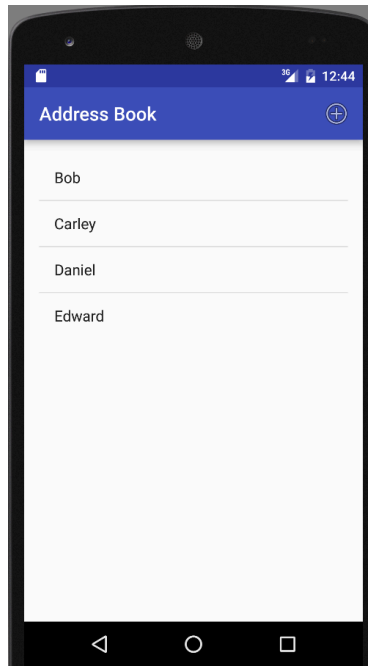
Delete Contact

When the user clicks on the Delete Contact option, the app should pop up an alert view to double check with the user:

If the user decides to delete the contact, then the contact should be removed from the first list view as a result.

At last, make sure that your app stores the contact information persistently so that the contacts remain in the address book when the app is launched again after the user clicks on either the Home or Back button.

Grading rubrics:

Function description	Points
The first list view showing contacts.	5
The add menu option on the first list view.	5
Add Contact view: editable (5 points), successfully save contact information and display it on the list view (5 points).	10
Touching an item on the list brings a view showing detailed contact information.	3
Edit and Delete options appear in the context menu.	2
Edit Contact view: editable (1 point), scrollable (1 point), and successfully save contact information and display it on the list view (3 points).	5
Delete Contact: 1) pop an alert view asking for the user's re-confirmation (2 points); and 2) contact is successfully deleted from the list (3 points).	5
The contact information is saved persistently.	5