Now that you have become more familiar with this application, it's time to add some new functionality.

In this assignment you will update the application to implement this new functionality using the updated UML class diagram as a guide.

Review criteria **less**

You will be graded on your understanding of the application code and design, your ability to translate from a UML class diagram to java code, and the correctness of your code.

Step-By-Step Assignment Instructions **less**

Setup instructions

Use this tutorial as your guide: Peer Review 4 Tutorial

The tutorial will begin by giving you the option to pick a code base:

Option 1: starter code base (**recommended especially for those without Android programming experience** -- you will only have to follow steps 1, 18, 19, and 20 of the tutorial)

SharingApp_contacts_starter.zip

Option 2: items only code base (you will have to follow all steps 1-20 of the tutorial)

SharingApp_items_only.zip

Use the following updated UML class diagram to assist you with your understanding and implementation:

UML Class Diagram Contacts.png

To help clarify what is expected in this version of the app, please view the updated user stories:

User Stories with Contacts.pdf

About the application

This version of the app should accommodate the new contacts feature. In particular:

- ContactsActivity should be accessible from the MainActivity.

menu.png

contacts_menu_item.png

- ContactsActivity should be implemented as a ListView.
- An owner should now be able to add a potential borrower (a contact) to their contacts. Each contact must have a unique username and an email.

ContactsActivity.png

AddContactActivity.png

- An owner can edit or delete a contact, but not if the contact is currently borrowing an item, i.e. the contact is a borrower.

EditContactActivity.png

- Owners are now required to select a contact to be the borrower of an item when changing the status of an item from “Available” to “Borrowed”. That is, it is no longer sufficient to enter the borrower’s username as a string -- now the borrower must be picked from the owner’s list of contacts.

EditItemActivity_available.png

EditItemActivity_select_borrower.png

EditItemActivity_borrowed.png

Submission Instructions

Part 1 Submission will test your ability to translate the UML class diagram into Java code. When you are ready to submit your code, include the following two files in a folder:

- **Contact.java**
- **ContactList.java**

Then compress the folder into a ZIP archive. Windows users can use 7zip or WinRAR. Upload it where prompted.

Part 2 Submission will test the correctness of your code

In order to grade your assignment, you will need to submit a 5 minutes or less demo video of your app that shows the following steps as a continuous interaction without crashing (if possible):

1. Start the video of your app from the MainActivity.
2. From the MainActivity, navigate to the ContactsActivity.
3. From the ContactsActivity, add a new contact to your contact list. Show that you can enter a username and email, save this contact to your contact list.
4. Show that by selecting (long clicking) a contact in the contact list, you can edit this contact. Update the email address of a contact.
5. Show that you can delete one of your contacts.
6. Go back to the MainActivity and look at your available Items.
7. Add an item to your inventory (if you don't already have an available item).
8. Edit an item in your inventory by long clicking on the item. Click the "Available" toggle. The toggle should now say "Borrowed" and a box should appear below to indicate the name of the borrower. By default the box will show the username of the first contact in your contacts. If you have more than one contact in your contacts you can click this box and then select the desired contact.
9. Finally, press "Save" to return to your inventory.

Upload your demo video to Internet Archive or YouTube and then provide a link to it where prompted.