

ITIS/ITCS 4180/5180 Mobile Application Development

Homework 2

Date Posted: 09/07/2015 at 22:00

Due Date: 09/14/2015 at 17:00

Basic Instructions:

1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
 - a. Only a single group member is required to submit on moodle for each group.
 - b. The file name is very important and should follow the following format:
Group#_HW02.zip
 - c. You should submit the assignment through Moodle: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

Homework 2 (100 Points)

In this assignment you will get familiar with Android Intents (Explicit/ Implicit), Data Passing between Intents (Parcelable/ Simple Data Passing), Starting activity for result and Alert Dialogs. This is a Contacts Directory App , where user gets to enter Contacts information, the name/ mobile number, along with other details. User should be able edit, display and delete Contacts. Main screen is displayed in Figure 1.

NOTE: This is a stand alone Contacts App, it has nothing to do with the System Contacts App

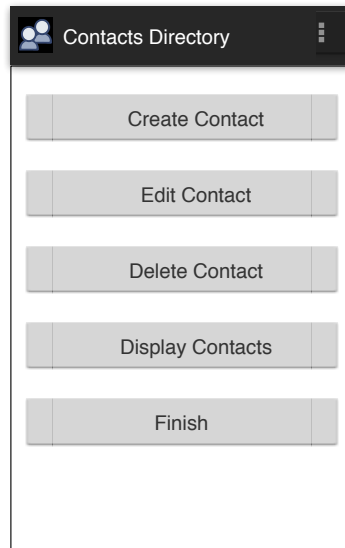


Figure 1, Main Activity

Main Activity (10 Points)

The Main Activity contains the 5 buttons arranged as displayed in Figure 1. The Main Activity should maintain a LinkedList of the current Contacts Objects, the LinkedList should be shared with the other activities through intents when needed. The activity requirements are as follows:

- The contact list should be stored in the Main Activity, you should use an ArrayList to store the list of Contact Objects. The list should be passed using intent extras to other activities when needed, you should not declare the list as static.
- The Create Contact, Edit Contact, Delete Contact, Display Contacts, and Finish buttons, should start the Create Contact Activity, Edit Contact Activity, Delete Contact Activity, Display Contacts Activity and finish the activity respectively.

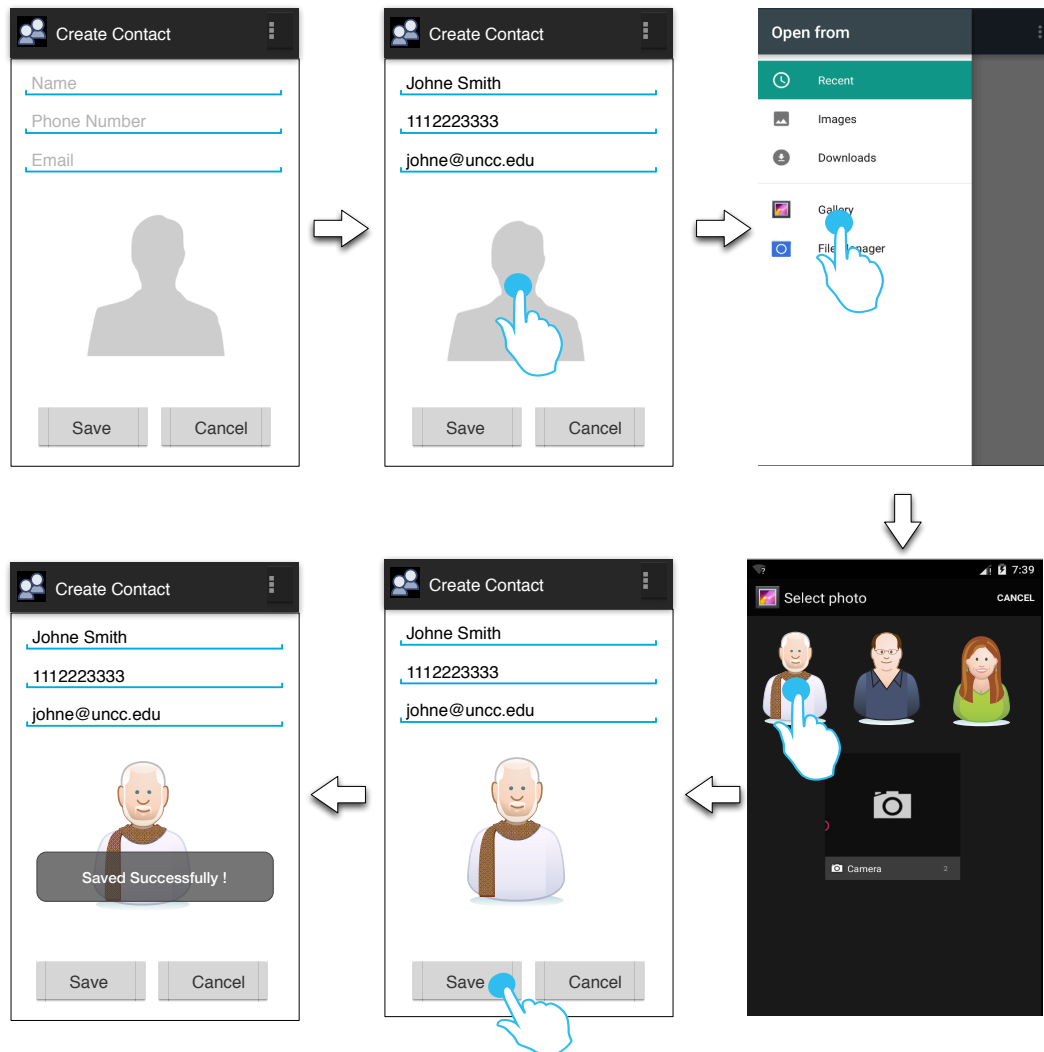


Figure 2, Create Contact Activity

Create Contact Activity (20 Points)

The Create Contact activity is displayed in Fig 2. Implement the following requirements:

- When the user selects “Create Contact” button in the Main Activity, the Create Contact Activity should be started as shown in Figure 2.
- The Contact Name should not exceed 50 characters. The Phone Number should be of 10 digits and the email should follow the correct email format. Any missing or incorrectly entered information the user should be alerted through Toast messages.
- Profile Photo should initially display a default avatar photo. The user has a choice to associate a photo with the contact. Tapping Profile Photo should enable user to get an image from the system Image Gallery App. This should start New Activity for result. The selected image should be returned back to the previous activity. Hint: for intent information check <https://developer.android.com/guide/components/intents-common.html>
- Clicking the Save button should save the fields as a Contact object, send it back as a result to the Main Activity so that it can be stored in the Contact list and finish the Create Contact Activity.

- Clicking the Cancel button should finish the Create contact activity.
- NOTE: To save the image selected, you should save the **Uri** object that points to the selected image, do not store the full image bitmap.



Figure 3, Edit Contact

Edit Contact Activity (25 Points)

The edit Contact activity is displayed in Figure 3. Implement the following requirements:

- When the user selects “Edit Contact” button in the Main Activity, the Edit Contact Activity should be started as shown in Figure 3.
- Clicking the Select Contact button should show an alert dialog displaying the list of contact names sorted in ascending order. The list enables the user to pick which

Contact to edit. Upon selecting a contact from the list, the Contact information should be populated in the Edit Contact activity as shown in Figure 3.

- The UI for this activity is similar to the Create Contact Activity and should follow the same text size and implementation and restrictions.
- Clicking the Save button should save the fields as a Contact object and send it back as a result to the Main Activity so that it can be updated and stored in the Contact list.
- Clicking the cancel button should finish the edit contact activity and does not update the edited contact.

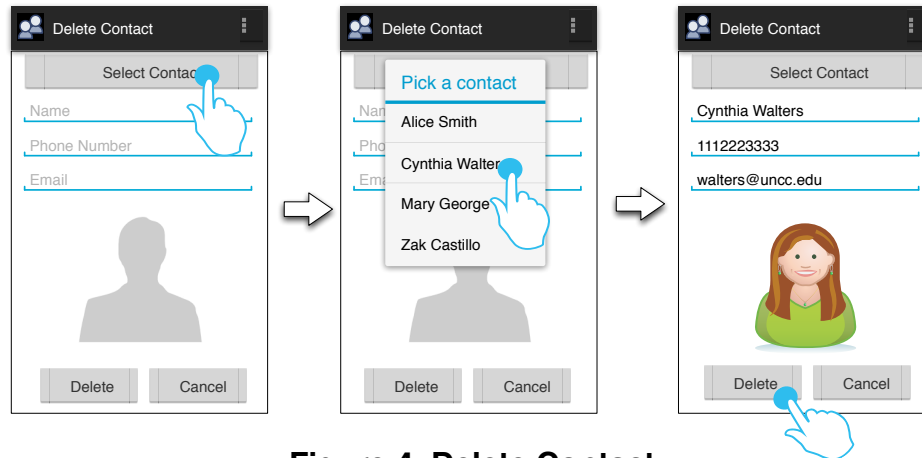


Figure 4, Delete Contact

Delete Contact Activity (25 Points)

The delete Contact activity is displayed in Figure 4. Implement the following requirements:

- When the user selects “Delete Contact” button in the Main Activity, the Delete Contact Activity should be started as shown in Figure 4.
- Clicking the Select Contact button should show an alert dialog displaying the list of contact names sorted in ascending order. The list enables the user to pick which contact to delete. Upon selecting a contact, the contact information should be populated in the Delete Contact activity as shown in Figure 4.
- All the EditText components and ImageViews should **not** be clickable, and should only be used to present the current Contact’s information.
- Clicking the Delete button should delete the current Contact and send the Main activity the required information in order to delete this Contact from the Contact list. Clicking the Cancel button should finish the Delete Contact activity and should not delete the Contact.



Figure 5, Display Contacts

Button	Action
⏮	Show the first contact
⏪	Show previous contact
⏩	Show next contact
⏭	Show last contact

Table 1, GUI Actions

Display Contacts Activity (20 Points)

The display Contacts activity is displayed in Figure 5. Implement the following requirements:

- When the user selects “Display Contacts” button in the Main Activity, the Display Contacts Activity should be started as shown in Figure 5.
- This activity enables the user to view all the currently stored Contacts.
- The actions required from the each of the GUI images are listed in Table 1. Upon clicking finish the activity should be finished.
- Tapping Email should open the default email app in the device with the selected email passed to the email app in the “To:” field, see Figure 6(a).
- Tapping the Phone Number, should initiate a call to the number, see Figure 6(b).

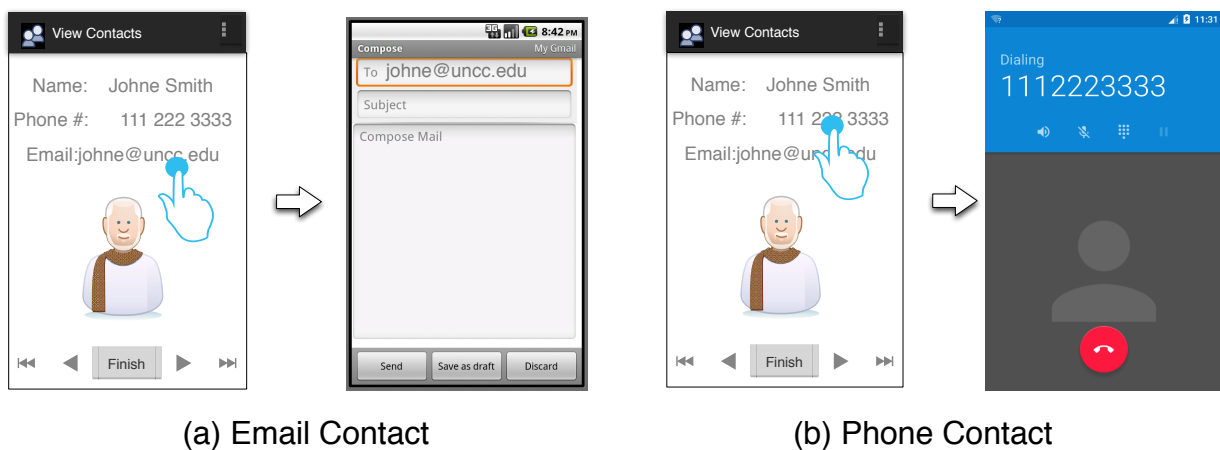


Figure 6, View Contacts