Mobile App Engineering and User Experience - Fall 2018

2nd Assignment – Contacts List

Published: Monday Oct. 8th, 2018

Deadline: Wednesday Oct. 24th 11:55am, 2018

In this assignment, you will be building an app to practice using fragment and multiple activities of an Android. The required API level for this assignment is 23 (Android 6.0).

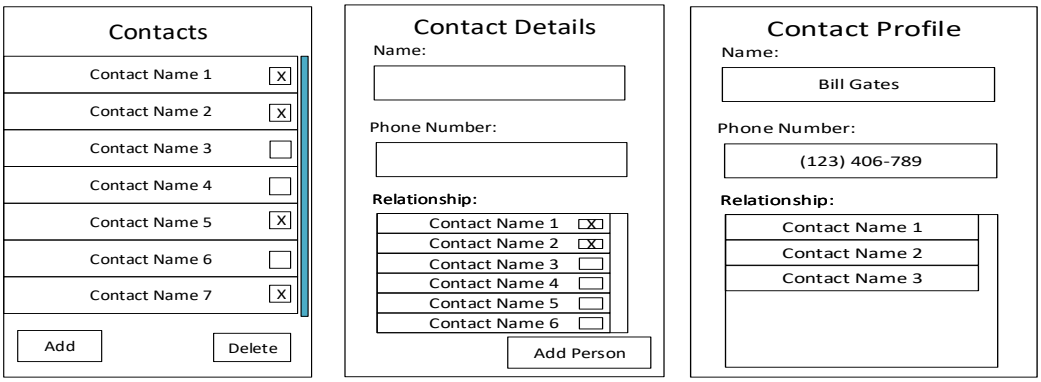
As before, if you do not have access to an Android phone, please contact John Scafidi as soon as possible. The phones are available today for daily loans near the campus. **Note that you are NOT allowed the root or do any other modifications to these phones. If you modify or root the phones, you bought them.**

All projects are required to be submitted with up to a one-page report that specifies what was implemented, and until what point range. You need to also justify your design decisions. Graduate students need to submit additional report of the graduate student requirements.

1) 60 points maximum

Write a contacts app in portrait view.

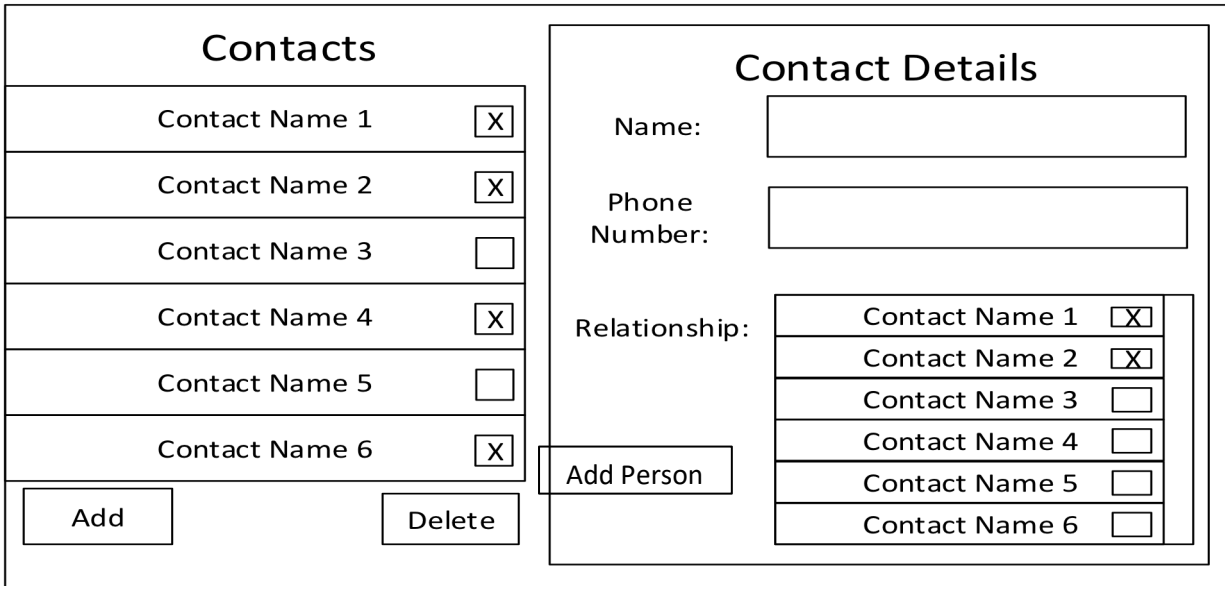
(1) There should be a scrollable list view of all contacts. (5 point);
(2) There should be a checkbox in each name. The user can multiple check the names, and press "delete" button to delete them all (5 points).
(3) When user click the contact name, it should display the "contact profile" (5 points).
(4) Use database to store the contact names (10 points).
(5) When user click the add button, it need to jump to the new activity "Contact Details" (5 points);
(6) In "Contact Details" activity, the user can input the name, phone number and relationship with contact that exists in the contacts list (5 points). User press "Add Person" button to confirm input information.
(7) The "relationship" describes the link between the different contact people. The list should display the whole list of contacts. And the checked name should be arranged to the front of the contacts list (10 points).
(8) When a user clicks the contact name under the "relationship" in "Contact Profile", it should jump to the contact profile of the clicked name by creating a new activity (5 points).
(9) When a person, for example, Emily is selected as related to person Bob. The both the relationship under their name should be checked (10 points).



2) 100 points maximum

The app needs to have a different layout with landscape view.

(1) Use fragment to display the contact list on the left (5 points);
(2) The information you input in portrait view or landscape view need to keep constant when the view changes (10 points).
(3) When user click the contact name, the contact profile should be shown on the right side (5 points).
(4) When user click the "Add" button, the "Contact Details" need to show on the right side (5 points).
(5) The delete button, relationship list are the same as portrait view (5 points).
(6) The left fragment and right fragment should occupy 2/5 and 3/5 of the width of screen, respectively (5 points).
(7) When a user clicks the contact name under the "relationship" in **"Contact Profiles",** it should jump to the contact profile of the clicked name. And the contact profile should also show on the right side of landscape view (5 points).



Graduate students' additional requirements (up to -30 for graduate students who do not complete this):

Add the ability to store a photo with each contact, which is displayed near the upper right corner of the contact details fragment. Make sure that photos are not lost when your app terminates (you are allowed to store photos as a file). When no photo is available, display a generic head icon in the same area. Tapping this item

should allow the user to take a photo and automatically assign it to the contact. If a photo is available, tapping this photo should lead to a zoom in animation until the photo occupies the entire screen. Do your own research in the Android Libraries to find the appropriate class for the zoom feature.

Upload your solution source code, apk package and report in the same zip or tar.gz file named as netid_yourname. The apk named as the name defined in manifest for the main activity. The apk file should be copied to the root directory of the submission folder, also keep the original apk in the bin folder.

Detailed submission instructions:

All of the submitted materials should be in one folder. The folder should be zipped and named

as netid_yourname.

In the folder, there should be three parts:

1. .apk file.

I will use this file to install your assignment in a test phone and check all of the requirements

2. 1-page report.

It should specify what was implemented, and until what point range. You need to also justify your design decisions. Graduate students need to submit an additional report of the graduate student requirements.

3. Source code folder.

This should be the entire folder of your assignment. If necessary, I will rebuild the app with your submitted code.

To test if your submitted source code is sufficient, you can try to load your assignment from this submission folder. You can open an existing Android Studio project following the steps below:

open “Android Studio” -> click “Open an existing Android Studio project” -> find the folder in your submission folder -> click “Open” -> Android Studio will load the project automatically -> Try to “Run” app in this new project -> If your project can be built successfully, it means the source code is sufficient enough for me to rebuild your project and app.

I listed an example list of this source codes folder should include. Please note that different project should have different names, your source code does not need to exactly follow the list below.

Assignment2-|

|- app

|- build.gradle

|- gradle

|- gradle.properties

|- gradlew

|- gradlew.bat

|- local.properties

|- settings.gradle

|- Assignment2.iml