ITIS/ITCS 4180/5180 Mobile Application Development
Homework 2
Date Posted: 09/14/2012 at 04:00pm
Due Date: 09/21/2012 at 11:59pm

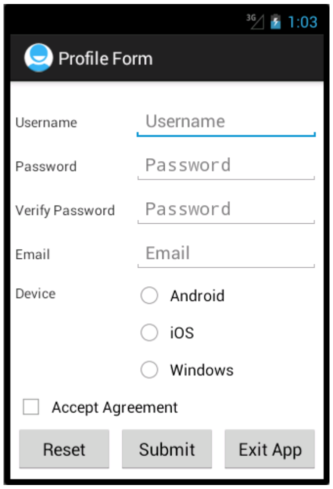
Basic Instructions:

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. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is HW#.zip
   b. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

Homework 2 (100 Points)

In this assignment you will build your first Android application. You will get familiar with common Android components and how to interact with them. You will build a form that accepts user input, verify the various input values, and show appropriate error messages when needed. This assignment consists of 3 parts:

Part 1 (35 Points): Building the Form (MainActivity)



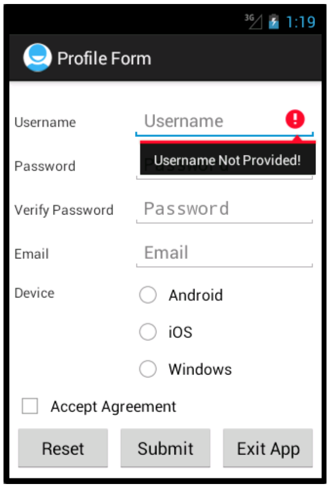
You will build a Profile Form application similar to the one shown in the Figure to the right. You will be dealing with the layout files, strings.xml, and drawable-ldpi files to create the form. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse. To build the form, you will perform the following tasks:

1. Use TextView components for the Username, Password, Verify Password, Email and Device as seen in the Figure. The actual string values for these labels must be read from the strings.xml file provided.
2. Use EditText components for the Username, Password, Verify Password, and Email Input fields. Each of the inputs will have a hint (placeholder) value read from the strings.xml file. For example, the Username input has a hint value equal to Username.
3. For the Device selections, use a RadioGroup with three RadioButton components. By default, the RadioButtons must be unselected. The labels for each radio button must be read from the strings.xml file programmatically (Do this in the onCreate() method).
4. Add a CheckBox with the label Accept Agreement (from strings.xml), which by default should be unchecked.
5. Finally, you will add 3 buttons, a Reset, Submit, and Exit App button. The text for these buttons must also be read from the strings.xml file.
    a. The Reset button should have a click event listener that will reset all form fields (Clear all input fields, and remove all selections for radio buttons check boxes).
    b. Clicking an enabled Submit button will verify all the provided input values and selections as seen in the following part (Part 2).
    c. The Exit App button will exit the application.

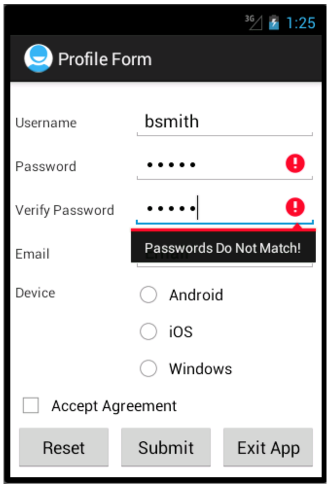
Part 2 (35 Points): Input Verification (MainActivity)

Once you build your Profile Form, users are able to input values and make various selections. It is now your responsibility to verify the inputs and selections once the user clicks the Submit button. When verification fails, you will be asked to show a proper error message to the user. Use the error message values provided in the strings.xml file. The form verification should be handled as follows:

1. If any of the Username, Password, or Email input fields are empty show the corresponding error message using the setError() method. For example, if the Username field is empty, you will show “Username Not Provided!” (read from the username_error_message string in strings.xml).
2. If the Password and the Verify Password input fields are not equal, show the provided error message using the setError() method.
3. If the email entered by the user is invalid show the provided message using the setError() method. To verify the email, use the Java Pattern and Matcher classes.
4. If no device is selected, show the provided error message in a Toast.
5. Users must check the Accept Agreement checkbox. If it is not checked then show a Toast message with the provided error message.
6. When all of the above cases are verified and all inputs are correctly inputed, upon clicking the submit button create an explicit intent to start the DataReview activity. All the information should be sent to the DataReview activity in a FormData class provided. Hint: You should investigate using the Serializable interface or Parcelable interface.



Username Not Provided!

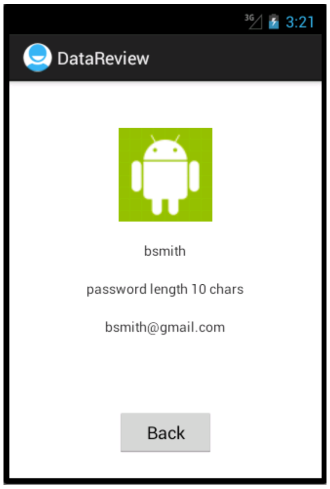


Username: bsmith

Passwords Do Not Match!

Part 3 (30 Points): Data Review (DataReviewActivity)

This activity is started by the main activity and it is passed the form data in a FormData instance. The activity should display the data in the form as indicated in the figure. You are provided with three figures to represent the different device choices and the activity should display the image corresponding to the selected device. The activity displays the username, password number of characters, and email. In addition it should display the image corresponding to the selected device, which in this case was android. Upon clicking the back button the activity should finish and go to the main activity.



bsmith

password length 10 chars

bsmith@gmail.com