Lab Assignment #2 – Developing an Android Shopping App

Due Date: 3-Jun-2018

Purpose: The purpose of this lab assignment is to:

- Use Android UI controls to develop an interactive mobile application

References: Textbook, ppt slides and Android tutorials (http://developer.android.com/guide/topics/ui/index.html). This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This assignment must be completed **individually** by all the students.
- You will have to **demonstrate your solution in a scheduled lab session** and upload the solution on e-Centennial through the assignment link.

Descriptions:

Students are asked to develop an Android application to allow the customers to place an order using their smart phone for the following descriptions. The main screen will display the company logo (think..., Best Buy like a real or imaginary business) and a button "Shop".

The second screen allows the customer to choose between categories of product list (Cell phones, Laptops or notebooks, Televisions or Smart TVs, Printers, and Accessories, etc). Use **a menu** to implement this list. The next screen displays the selection of products' brands (iPhone, Samsung, Black Berry, Google, LG etc) from the selected category. Use a **radio button** control to implement this selection.

After selecting the product brand, the user will be prompted with products (Samsung Galaxy S9+, Samsung Galaxy 9, Samsung Galaxy note 8, Samsung Galaxy 8+ and Samsung Galaxy 8) belong to the product brand Samsung similar with other products (iPhone X, iPhone 8 Plus, iPhone 8, iPhone 7 Plus and iPhone 7) belong to the product brand (iPhone) etc. Use check boxes to allow the user to order more than a product. (example: Samsung Galaxy S9+ and Samsung Galaxy 8)

The customer screen prompts the user to provide customer information. Use *EditText* controls and other UI elements to allow the user to enter **customer' information:** *name*, address, *credit card number*. The rest of the fields will be **different for each student**. For example, you may create fields phone number, delivery date, user name, password etc. Create 2-4 fields named as mentioned above. Provide **validation** for these entries using the proper methods/constructor for each UI control.

Use *TextView* objects to create the label components for your UI. Allow the user to use "**Back**" key to go back to the previous screen. Display the order information on the check-out screen when the user clicks on a button titled "**Place an Order"**

Use styles and themes to create a nice look and feel of your app. Use drawable objects to display the logo for the company, product category, product brands and products.

Evaluation Rubrics:

Criteria	Not Satisfactory 0-1	Satisfactory 2	Very Good 3	Excellent 4
Understanding specification and Code analysis	The code produces incorrect results due to poor understanding.	The code produces correct results but does not display them correctly due to incomplete analysis.	The code works and meets most of the other specifications. Very good code analysis.	The code works and meets all the specifications and well thought out code analysis.
Layout Design and Resources file.	Poor design and no resources files used.	Good design with fair use of resources file.	Very good design with appropriate use of resources file like string.xml.	Excellent design with excellent use of resources files.
Coding Standards	No name, student number, Poor use of white space, Poor use of variables.	Includes name, and student number, White space makes program fairly easy to read. Good use of variables.	Includes name, and student number, Good use of white space. Very good use of variables.	Includes name, and student number, Excellent use of white space. Excellent use of variables.
Code debugging and Demonstration / Screen shots	Unable to debug and run program due to the syntax and logical errors and no testing is done.	Able to run program but have few logic errors and some testing is done.	Able to debug and run program correctly without any logic errors and Thorough testing is completed	Able to run program without logical errors. Thorough and comprehensive testing is completed
Program Delivery	The assignment is completed (50-70%) and submitted within three days late with in correct format (.zip file format)	The assignment is completed (70-80%) and submitted within two days late with correct format (.zip file format)	The assignment is completed (80-90%) and submitted within a 24 hours late with correct format (.zip file format)	The assignment is fully completed (100%) and submitted on time with correct format (.zip file format)
Grade	0-5	10	15	20

Android Apps - project Naming rules:

You must name your Android Studio application according to the following rule:

YouName_COMP304_Assignment2

JohnSmith_COMP304_Assignment2

Submission rules:

Submit your applications as **zip files with your android project and screen shots** that are named according to the following rule:

YourName_StudentNumber_AssignmentNumber.zip

Example: **JohnSmith_999999999_Assignment2.zip**

Academic honesty (Plagiarism and cheating)

All students must follow the academic honesty policies regarding Plagiarism and cheating on assignments, Quizzes or Tests. Centennial college's Academic Policy will be strictly enforced. To support academic honesty at Centennial College, all academic work submitted by students may be reviewed for authenticity and originality, with utilizing software tools.

For more details, please visit the Academic Honesty site on https://www.centennialcollege.ca/mycentennial/your-support/academic-support/student-academic-advising/academic-honesty/