

## **Android - Assignment # 3 Cruddy Pizza – individual**

### **Overview:**

This Cruddy Pizza assignment is primarily intended for students to demonstrate their knowledge of:

- Data persistence using SQLite (Order Info).
- Data persistence using Shared Preferences (Bilingual UI Settings).
- String Arrays
- UI design and internationalization

### **Functionality:**

This application will display a list of options for building a pizza. Once the order is finished it is to be saved (created). Additional functionality will allow the user to display all orders (read), a specific order (read), modify the order (update), and cancel the order (delete). Let the user know when each operation is complete. The UI is to be able to be switched between English and French and maintain state on that setting between start-ups.

### **Layout Options:**

Layout is up to the student and should look professional with a good choice of colors / images and have a good functional layout with appropriate control types.

### **Data persistence – SQLite:**

Implement the use of an SQLite database into your application in order to demonstrate your familiarity with that technology. Your solution must have full CRUD capability. Orders may be “atomic” in that you do not need to keep a separate table of customers for future orders or marketing data. Validation is required for all non-selectable fields e.g. phone number.

The following order information must be retained:

- Size – small, med, large, extra large
- Toppings – max 6
- Order date
- Order time
- Customer Name (may be a single field)
- Customer Address (may be a single field)
- Customer Phone Number

### **Getting Started/Recommended Steps:**

1. Plan out the functionality of your applications and the design of your UI.
2. Gather all the images you intent to use in your UI.
3. Create a new project and add an appropriate custom image and application name.
4. Store any images required for your UI in the application.
5. Store any strings you intend to use in your application (both English and French).
6. Build out your UI.
7. Create a database and copy over any DB code from in-class demonstrations you may need. Modify this code to fit your new application, remembering to include exception handling for ALL CRUD operations.
8. Test your application on different sized/density virtual devices (in both portrait and landscape mode) to demonstrate that it always looks good and is working correctly. Make sure it does not crash when the orientation changes

