PROJECT SPECIFICATION

# Developing First Android App

## Code Quality

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Correctly use ViewModel and LiveData lifecycle classes in an Android app | The Detail screen needs to add the new item to the view model. The listing screen should be listening to that model and show the new item. |
| Correctly implement Single Activity architecture | There should only be one activity: MainActivity. Each screen should be a fragment. |
| Write error-free code | The project's code is error-free. |

## Layouts

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | |
| Create layouts using the correct ViewGroups and Views in an Android app. | The project correctly implements LinearLayout and ConstraintLayout to match the complexity of the layout of a page. Using code comments, the project justifies the use of ConstraintLayout or LinearLayout |
| Apply Databinding in Layouts to show the correct data to users in multiple layouts. | 1. All layouts will use the <layout> tag to support Databinding.<br>2. Detail screen uses the <data> element.<br>3. Databinding is set to the appropriate setting in the app build.gradle file |
| Correctly use the <data> and <variable> elements within the layout. | The detail layout contains an <data> element with the name of the variable and the class associated with it.<br><br>• All EditViews correctly refer to created class variable |
| Create a multi-screened Android Application | • The app contains at least 5 screens.<br>• The app contains correctly laid-out labels and edit fields for each screen.<br>• The app contains button positioned below the text fields |

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| using Android widgets. | |
| | |
| List screen uses ScrollView and LinearLayout for showing a list of items and one Floating Action button for going to the detail screen. Creates a layout for the item. | 1. A new item layout is created for each item<br>2. New item layout is added to LinearLayout<br>3. Layout is updated with items added on the detail screen |
| Create a detail screen that shows two columns of labels and edit views to enter in a new item. | 1. Layout created with a label & edit view for each item<br>2. Uses data binding to save data<br>3. Uses a save button to save data to view model |

## Navigation

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | |

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| | |
| Create a navigation file that correctly takes a user from one page to the next in an Android app | The app needs to traverse the following screens in the correct order:<br><br>• Login<br>• Welcome<br>• Instructions screen<br>• Listing screen<br>• Detail screens<br>The app should also be able to navigate back via the back arrow or the back button.<br>• A navigation file has been created that defines a start destination.<br>• All destinations have a fragment, label and action associated with it. |
| Use Databinding for click listeners on a navigation screen in an Android app. | 1. All code will use the DataBindingUtil class to inflate the layout.<br>2. All click listeners are connected via the DataBindingUtil class and uses the NavController to navigate to the next screen. |
| Create a Logout menu to return to the Login screen. | This menu will appear only on the Shoe Listing screen and will return the user to the login screen |

## Suggestions to Make Your Project Stand Out!

1. The student uses styling on the TextViews and buttons
2. Layouts look nicer than the basic layout
3. Navigation uses Capitalized names for the labels
4. Navigation uses enter/exit animations