Review-Corrections/Implementations

Regular Text: Comments specified by Team 5
Colored Text: Code review implementation/ code errors fixed

1. BAD PRACTICE
Method might ignore exception (multiple occurrences)

BookInfo.java @ line 47, 93

**CODE SNIPPET:**
catch (Exception ex)
{
    /* Display the error */
}

**COMMENT:**
As a test case we switched the internet off and the code was unable to catch the exception.

- Method names should start with a lower case letter (7 occurrences)

    BookInfo.GetBitMapImage()
    BookInfo.GetBookDetailsByISBN(String)
    BookInfo.GetHoldingInfoByISBN(String)
    BookInfo.GetISBNValueByImage()
    ScanBarcode.SetBitMapImageByID(int)
    Service.GetDataFromServer(URL)
    Utility. GetParsedData(List, String)

**COMMENT:**
It is considered a good programming practice and reduces the confusion between the classes and the methods.

**Correction:** We have renamed the above method names with camel case convention.

- Class names should start with an upper case letter (5 occurrences)

    scanbarcode.R$attr
    scanbarcode.R$drawable
    scanbarcode.R$id
    scanbarcode.R$layout
    scanbarcode.R$string

**COMMENT:**
The class names are auto generated and therefore it is a false warning.

**Correction:** No correction

2. CORRECTNESS

- Call to equals() with null argument (2 occurrences)

**Correction:** It has been fixed

Call to equals() with null argument in BookInfo.GetHoldingInfoByISBN(String) @lines: 77 - 96

**Correction:** It has been fixed
CODE SNIPPET:

```java
public Map<String, String> GetHoldingInfoByISBN(String isbnValue)
{
    String result = null;
    try {
        /* Define the URL we want to load data from. */
        URL myURL = new URL("http://www.openmarun.org/readhtml.php?isbnValue=" + isbnValue);
        result = service.GetDataFromServer(myURL);
        if (!result.equals(null)) {
            String infoLists[] = result.split(";");
            for (String info : infoLists) {
                String infoList[] = info.split(":");
                bookHoldDetails.put(infoList[0].trim(), infoList[1].trim());
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return bookHoldDetails;
}
```

**COMMENT:**
This method calls `equals(Object)`, passing a null value as the argument. According to the contract of the `equals()` method, this call should always return false. So `result.equals(null)` will always return false so the if condition would return true in all the cases. If `result` is null, it will generate a `NullPointerException`.

**Correction:** It has been fixed: Changed based on URL specified.

Call to equals() with null argument in Utility.GetParsedData(List, String) @lines: 17 - 50

CODE SNIPPET:

```java
{String attrName = parser.getAttributeName(counter);
if(!attrName.equals(null) && dataKeyList.contains(attrName))
{ parsedDataList.put(attrName, parser.getAttributeValue(counter)); } }
```  

**COMMENT:**
This method calls `equals(Object)`, passing a null value as the argument. According to the contract of the `equals()` method, this call should always return false. According to the JavaDoc, the method `getAttributeName(counter)` will never return a null value. So `attrName.equals(null)` will always return false so the if condition would return true in all the cases. If `result` is null, it will generate a `NullPointerException`.

**Correction:** It has been fixed: Changed based on URL specified.

3. MALICIOUS CODE VULNERABILITY (24 occurrences)

The team has included the Zxing API from Google which is the prime cause for this bug which could not be verified without the source code.

**Correction:** It has been fixed: No changes necessary.
4. PERFORMANCE (7 occurrences)

- Inefficient Map Iterator
  Correction: It has been fixed: Changed based on URL specified.

ScanBarcode$1.onClick(View) @ lines: 63 - 109

**CODE SNIPPET:**

```
………………………………………………
for (String key : bookHoldDetails.keySet())
{ builder.append("n" + key + ": " + bookHoldDetails.get(key)); } }
```

**COMMENT:**
This method accesses the value of a Map entry, using a key that was retrieved from a `keySet` iterator. It is more efficient to use an iterator on the `entrySet` of the map, to avoid the `Map.get(key)` lookup. Below is the link that gives the usage of `entrySet`.

http://leepoint.net/notes-java/data/collections/maps/map_iteration.html

- Method invokes inefficient Number constructor; use static value Of instead (6 occurrences)
  Correction: Fixed, made necessary alterations.

The team has included the Zxing API which is the prime cause for this bug which could not be verified without the source code.

**SUGGESTIONS:**
- In the current version of the application, the image file name is hard coded and it takes only one image at a time as the input. It would be good to have an alternative way to switch static barcode images in real time simulating the use of camera.
  Correction: Real application would activate the mobile camera and would process the image stored in the folder. However, emulator does not support webcam/camera. Rather static image is an good option.

- With more comments, code readability will increase. For example, the comments should include the method functionalities, control flow, possible exceptions etc.
  Correction: Implemented these ideas.