

Static Testing Documentation

Project: Electronic Store

Course: SWE 303

Worked : Antea Koxherri

1. Introduction

Static testing was performed in order to analyze the source code without executing the application. The goal of this phase was to identify potential bugs, bad coding practices, and maintainability issues early in the development process.

2. Tools Used

2.1 SpotBugs

SpotBugs was used to statically analyze the Java bytecode and detect potential runtime errors such as concurrency issues, bad practices, and possible bugs that could appear during execution.

2.2 SonarQube

SonarQube was used to analyze the source code quality by detecting bugs, code smells, deprecated usages, duplicated code, and possible null pointer risks. It also helps improve readability and maintainability of the code.

3. SpotBugs Analysis Results

The screenshot shows an IDE interface with several panes:

- Package Explorer:** Shows the project structure for "electronic-store". It includes packages like com.electronicstore, com.electronicstore.controller, com.electronicstore.model.inventory, com.electronicstore.model.sales, com.electronicstore.model.users, and com.electronicstore.model.utils.
- Project Explorer:** Shows the same project structure.
- BillingCont... App.java**: A code editor window showing Java code for an Application class.
- Bug Explorer:** This is the active pane, showing a single detected issue:
 - Severity: Scary (1)
 - Confidence: Normal confidence (1)
 - Description: Instance-getter method of class using singleton design pattern is not synchronized. (1)
 - Details: Instance-getter method of class using singleton design pattern (com.electronicstore.model.utils.SessionState) is not synchronized.
- Other Panes:** Javadoc, Problems, Console, SonarQube Report, Bug Info, SonarQube Issue Locations.

After running SpotBugs on the project, one main issue was detected with normal confidence severity.

Detected Issue:

- **Instance-getter method of a Singleton class is not synchronized**

Description:

SpotBugs reported that the getInstance() method of the Singleton class SessionState (com.electronicstore.model.utils.SessionState) is not synchronized. This may cause issues in a multithreaded environment, where more than one instance of the class could be created.

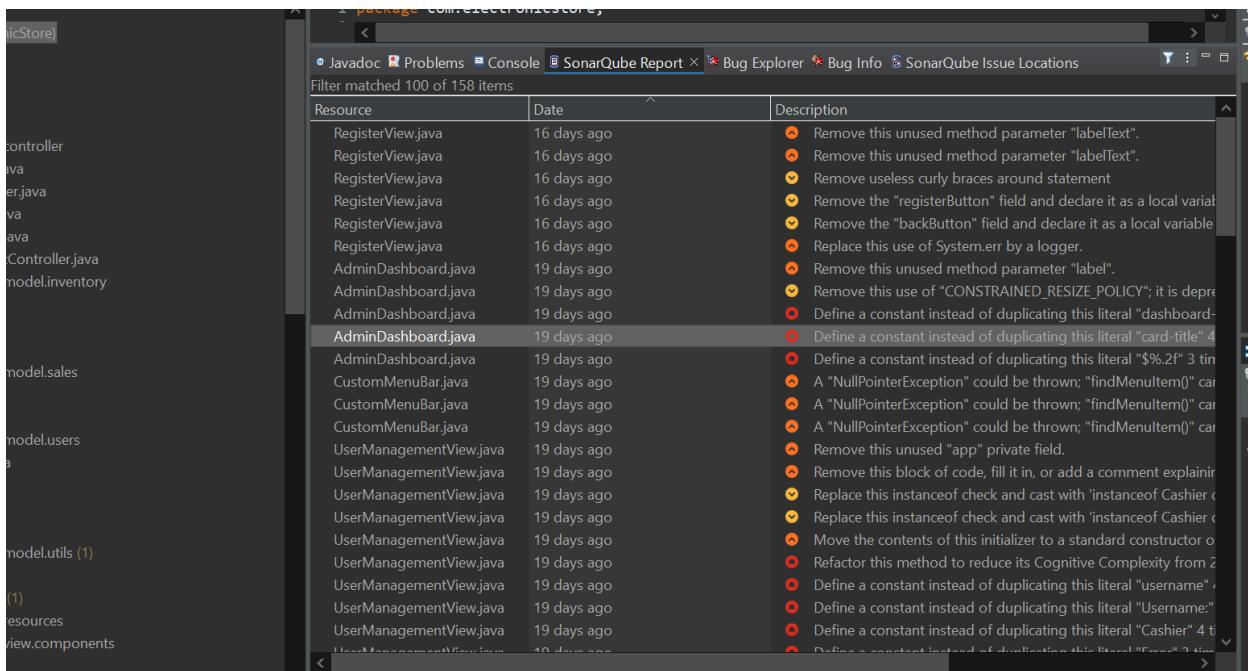
Severity: Normal confidence

Type: Design issue

Action Taken:

This issue was documented but not fixed, as the current application is not designed to run in a multi-threaded environment. However, if the application is extended in the future, synchronization should be added.

4. SonarQube Analysis Results



The screenshot shows the SonarQube Report view within an IDE. The left sidebar displays package structures for controller, model.inventory, model.sales, model.users, and model.utils. The main pane lists 100+ issues, each with a resource name, date, and a detailed description of the detected problem. Some issues are marked with orange icons, while others are marked with red or green.

Resource	Date	Description
RegisterView.java	16 days ago	Remove this unused method parameter "labelText".
RegisterView.java	16 days ago	Remove this unused method parameter "labelText".
RegisterView.java	16 days ago	Remove useless curly braces around statement
RegisterView.java	16 days ago	Remove the "registerButton" field and declare it as a local variable
RegisterView.java	16 days ago	Remove the "backButton" field and declare it as a local variable
RegisterView.java	16 days ago	Replace this use of System.err by a logger.
AdminDashboard.java	19 days ago	Remove this unused method parameter "label".
AdminDashboard.java	19 days ago	Remove this use of "CONSTRAINED_RESIZE_POLICY"; it is deprecated
AdminDashboard.java	19 days ago	Define a constant instead of duplicating this literal "dashboard"
AdminDashboard.java	19 days ago	Define a constant instead of duplicating this literal "card-title" 4 times
AdminDashboard.java	19 days ago	Define a constant instead of duplicating this literal "%2f" 3 times
CustomMenuBar.java	19 days ago	A "NullPointerException" could be thrown; "findMenuItem()" can return null
CustomMenuBar.java	19 days ago	A "NullPointerException" could be thrown; "findMenuItem()" can return null
UserManagementView.java	19 days ago	Remove this unused "app" private field.
UserManagementView.java	19 days ago	Remove this block of code, fill it in, or add a comment explaining why it is there
UserManagementView.java	19 days ago	Replace this instanceof check and cast with 'instanceof Cashier' or 'instanceof Manager'
UserManagementView.java	19 days ago	Replace this instanceof check and cast with 'instanceof Cashier or Manager'
UserManagementView.java	19 days ago	Move the contents of this initializer to a standard constructor or factory method
UserManagementView.java	19 days ago	Refactor this method to reduce its Cognitive Complexity from 2 to 1
UserManagementView.java	19 days ago	Define a constant instead of duplicating this literal "username"
UserManagementView.java	19 days ago	Define a constant instead of duplicating this literal "Username"
UserManagementView.java	19 days ago	Define a constant instead of duplicating this literal "Cashier" 4 times
UserManagementView.java	19 days ago	Define a constant instead of duplicating this literal "Employee" 4 times

SonarQube detected several issues across different classes of the project. These issues were mainly related to **code smells**, **unused code**, and **potential runtime problems**.

Examples of Detected Issues:

- Unused method parameters (e.g. labelText, label)
- Unused fields that could be converted into local variables
- Use of deprecated APIs (e.g. CONSTRAINED_RESIZE_POLICY)
- Duplicated string literals that should be replaced with constants
- Possible NullPointerException where methods may return null
- Use of System.out instead of a logging mechanism
- Unnecessary curly braces around statements

5. Fixing some of the Detected Issues

1.



This screenshot shows a specific issue from the SonarQube report. It highlights two occurrences of the same problem in the files Administrator.java and Manager.java. Both instances are marked with a green icon and the same fix: "Replace this use of System.out by a logger."

Administrator.java	few seconds ago	Replace this use of System.out by a logger.
Manager.java	few seconds ago	Replace this use of System.out by a logger.

```
7  *      @Override  
8  *      public void logout() {  
9  *          LOGGER.info("Manager logged out: " + this.getUsername());  
0  *      }  
1
```

2.

Administrator.java 13 minutes ago

Replace this use of System.out by a logger.

```
@Override  
public void logout() {  
    LOGGER.info("Administrator logged out: " + this.getUsername());  
}
```

3.

FileHandler.java	15 minutes ago	Remove this hard-coded path-delimiter.
FileHandler.java	15 minutes ago	Remove this hard-coded path-delimiter.
FileHandler.java	15 minutes ago	Replace this use of System.out by a logger.
FileHandler.java	15 minutes ago	Replace this use of System.out by a logger.
FileHandler.java	15 minutes ago	Remove this hard-coded path-delimiter.
FileHandler.java	15 minutes ago	Remove this hard-coded path-delimiter.
FileHandler.java	15 minutes ago	Remove this hard-coded path-delimiter.

```
import java.nio.file.*;  
import java.util.List;  
import java.util.ArrayList;  
  
public class FileHandler {  
    public static final String DATA_DIRECTORY = "store_data";  
  
    // Avoid hard-coded delimiters by using Path/resolve  
    private static final Path DATA_PATH = Paths.get(DATA_DIRECTORY);  
    private static final Path BILLS_PATH = DATA_PATH.resolve("bills");  
  
    static {  
        try {  
            // Create necessary directories  
            Files.createDirectories(DATA_PATH);  
            Files.createDirectories(BILLS_PATH);  
        } catch (IOException e) {
```

4.

UserManagementView.java 17 minutes ago

Remove this unused "app" private field.

5.

ManagerDashboard.java 24 minutes ago

This block of commented-out lines of code should be removed. [+1 location]

```
private void updateSalesChart() {  
    if (view.getStartDate().getValue() != null && view.getEndDate().getValue() != null) {  
        |  
    }  
}
```

7.

Administrator.java	25 minutes ago	Replace this lambda with method reference 'Manager.class::isInstance'.
Administrator.java	25 minutes ago	Replace this lambda with method reference 'Cashier.class::isInstance'.

Before :

```
public String viewSystemStats() {
    int totalUsers = allUsers.size();
    int cashiers = (int) allUsers.stream().filter(u -> u instanceof Cashier).count();
    int managers = (int) allUsers.stream().filter(u -> u instanceof Manager).count();

    return String.format("")
```

After Fixing :

```
public String viewSystemStats() {
    int totalUsers = allUsers.size();
    int cashiers = (int) allUsers.stream().filter(Cashier.class::isInstance).count();
    int managers = (int) allUsers.stream().filter(Manager.class::isInstance).count();

    return String.format("")
```

8.

CashierDashboard.java	31 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
CashierDashboard.java	31 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
CashierDashboard.java	31 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
CashierDashboard.java	31 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
CashierDashboard.java	31 minutes ago	Remove this unused method parameter "label".

Before :

```
billCountText.setText(String.format("Bills Generated\n%d", billCount));
totalSalesText.setText(String.format("Total Sales\n$%.2f", totalSales));
averageTicketText.setText(String.format("Average Ticket\n$%.2f", averageTicket));
peakTimeText.setText(String.format("Peak Time\n%s", peakTime));
```

```
private Text createMetricText(String label) {
    Text text = new Text();
    text.getStyleClass().add("metric-text");
    text.setStyle("-fx-font-size: 14px;");
    return text;
}
```

After Fixing :

```
billCountText.setText(String.format("Bills Generated%n%d", billCount));
totalSalesText.setText(String.format("Total Sales%n$%.2f", totalSales));
averageTicketText.setText(String.format("Average Ticket%n$%.2f", averageTicket));
peakTimeText.setText(String.format("Peak Time%n%s", peakTime));
```

```
private Text createMetricText( ) {
    Text text = new Text();
    text.getStyleClass().add("metric-text");
    text.setStyle("-fx-font-size: 14px;");
    return text;
}
```

9.

SalesChart.java 37 minutes ago A "NullPointerException" could be thrown; "currentPeriod" is nullable here. [+2 locations]

Before:

```
if ((1 currentPeriod == null || currentPeriod.isEmpty()) &&
    (previousPeriod == null || previousPeriod.isEmpty())) {
    showNoDataMessage("No sales data available for comparison");
    return;
}

barChart.getData().clear();

'currentPeriod' is dereferenced.
if (!2 currentPeriod.isEmpty()) {
    XYChart.Series<String, Number> currentSeries = new XYChart.Series<>();
    currentSeries.setName("Current Period");
    currentPeriod.forEach((date, amount) ->
        currentSeries.getData().add(new XYChart.Data<>(date, amount))
    );
    barChart.getData().add(currentSeries);
}
```

After Fixing :

```
public void compareSales(Map<String, Double> currentPeriod, Map<String, Double> previousPeriod) {
    boolean currentEmpty = (currentPeriod == null || currentPeriod.isEmpty());
    boolean previousEmpty = (previousPeriod == null || previousPeriod.isEmpty());

    if (currentEmpty && previousEmpty) {
        showNoDataMessage("No sales data available for comparison");
        return;
    }
}
```

10.

AdminDashboard.java 44 minutes ago Define a constant instead of duplicating this literal "dashboard-card" 3 times. [+3 locations]
AdminDashboard.java 44 minutes ago Define a constant instead of duplicating this literal "card-title" 4 times. [+4 locations]
AdminDashboard.java 44 minutes ago Define a constant instead of duplicating this literal "%\$.2f" 3 times. [+3 locations]

Before:

```
card.getStyleClass().add("dashboard-card");
```

After Fixing :

```
private static final String STYLE_DASHBOARD_CARD = "dashboard-card";
private static final String STYLE_CARD_TITLE = "card-title";
private static final String MONEY_FORMAT = "%$.2f";
```

11.

SupplierManagementDialog.java	51 minutes ago	Remove this unused import 'javafx.beans.property.SimpleIntegerProperty'.
SupplierManagementDialog.java	51 minutes ago	Remove this unused import 'com.electronicstore.model.inventory.Category'.
SupplierManagementDialog.java	51 minutes ago	Remove this unused method parameter "supplier".
SupplierManagementDialog.java	51 minutes ago	Remove this unused method parameter "supplier".

Before :

```
private void handleEditSupplier(Supplier supplier) {  
    //  
}  
  
private void handleDeleteSupplier(Supplier supplier) {  
    if (AlertDialog.showConfirmation("Confirm Delete",  
        "Are you sure you want to delete this supplier?")) {  
        // Add delete functionality in controller  
        refreshTable();  
    }  
}
```

After Fixing :

```
private void handleEditSupplier() {  
    Supplier supplier = supplierTable.getSelectionModel().getSelectedItem();  
    if (supplier == null) {  
        AlertDialog.showError("No Selection", "Please select a supplier to edit.");  
        return;  
    }  
    // Implement edit functionality  
}  
  
private void handleDeleteSupplier() {  
    Supplier supplier = supplierTable.getSelectionModel().getSelectedItem();  
    if (supplier == null) {  
        AlertDialog.showError("No Selection", "Please select a supplier to delete.");  
        return;  
    }
```

12.

Bill.java	58 minutes ago	Remove this unused import 'java.util.stream.Collectors'.
Bill.java	58 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
Bill.java	58 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
Bill.java	58 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.
Bill.java	58 minutes ago	%n should be used in place of \n to produce the platform-specific line separator.

Before :

```
sb.append("-----\n");  
sb.append(String.format("Total Amount: $%.2f\n", totalAmount));  
sb.append("=====*\n");  
sb.append("Thank you for shopping with us!\n");
```

After Fixing :

```
sb.append(String.format("Total Amount: $%.2f%n", totalAmount));  
sb.append("=====*\n");  
sb.append("Thank you for shopping with us!%n");  
  
return sb.toString();
```

13.

App.java	1 hour ago	✖ Remove this unused "MANAGER_DASHBOARD_FXML" private field.
App.java	1 hour ago	✖ Remove this unused import 'java.io.IOException'.
App.java	1 hour ago	✖ Remove this unused import 'javafx.fxml.FXMLLoader'.
App.java	1 hour ago	✖ Remove this unused import 'com.electronicstore.view.components.AlertDialog'.

Removed the unused imports.

14.

ReportController.java 1 hour ago Remove this duplicated import.

Removed the duplicated import.

15.

Before :

```
    report.append(String.format("\nDate: %s\n", date_format(DateFormat)));
    report.append(String.format("Revenue: %.2f\n", dayRevenue));
    report.append(String.format("Cost: %.2f\n", dayCost));
    report.append(String.format("Profit: %.2f\n", dayProfit));
});
```

After

Fixing:

```
    report.append(String.format("%nDate: %s%n", date.format(DATE_FORMATTER)));
    report.append(String.format("Revenue: $%.2f%n", dayRevenue));
    report.append(String.format("Cost: $%.2f%n", dayCost));
    report.append(String.format("Profit: $%.2f%n", dayProfit));
});
```

The replacement of hardcoded newline characters (\n) with platform-independent line separators (%n and System.lineSeparator()) significantly improved code portability and resulted in a reduction of 35 SonarQube issues.

16.

Resource	Date	Description
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "username" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Username:" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Cashier" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Error" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "email" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Phone:" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Sector:" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Role:" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Email:" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "button" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "phone" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "button-small" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "sector" 4 times. [+4 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Name:" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Success" 3 times. [+3 locations]
UserManagementView.java	1 hour ago	● Define a constant instead of duplicating this literal "Manager" 3 times. [+3 locations]

Part of the code before fixing :

```
dialog.setResultConverter(buttonType -> {
    if (buttonType == ButtonType.OK) {
        Map<String, String> results = new HashMap<>();
        results.put("username", username.getText());
        results.put("name", name.getText());
        results.put("email", email.getText());
        results.put("phone", phone.getText());
        results.put("password", password.getText());
        results.put("role", role.getValue());
        results.put("sector", sector.getText());
        return results;
    }
    return null;
});

dialog.showAndWait().ifPresent(results -> {
    if (userController.adduser(
        results.get("username"),
        results.get("password"),
        results.get("name"),
        results.get("email"),
        results.get("phone"),
        results.get("role"),
        results.get("sector"))) {
        AlertDialog.showInfo("Success", "User added successfully.");
        loadusers();
    } else {
        AlertDialog.showError("Error", "Failed to add user.");
    }
});
```

After Fixing :

```

private static final String ROLE_ALL = "All";
private static final String ROLE_CASHIER = "Cashier";
private static final String ROLE_MANAGER = "Manager";
private static final String ROLE_ADMIN = "Administrator";

private static final String LABEL_USERNAME = "Username:";
private static final String LABEL_NAME = "Name:";
private static final String LABEL_EMAIL = "Email:";
private static final String LABEL_PHONE = "Phone:";
private static final String LABEL_ROLE = "Role:";
private static final String LABEL_SECTOR = "Sector:";
private static final String LABEL_STATUS = "Status:";

private static final String KEY_USERNAME = "username";
private static final String KEY_PASSWORD = "password";
private static final String KEY_NAME = "name";
private static final String KEY_EMAIL = "email";
private static final String KEY_PHONE = "phone";
private static final String KEY_ROLE = "role";
private static final String KEY_SECTOR = "sector";

private static final String MSG_SUCCESS = "Success";
private static final String MSG_ERROR = "Error";
private static final String MSG_USER_ADDED = "User added successfully.";
private static final String MSG_USER_ADD_FAILED = "Failed to add user.";
private static final String MSG_USER_UPDATED = "User updated successfully.";
private static final String MSG_USER_UPDATE_FAILED = "Failed to update user.";

private static final String CSS_BUTTON = "button";
private static final String CSS_BUTTON_SMALL = "button-small";

```

17.

UserManagementController.java 1 hour ago

Replace this use of System.err by a logger.

Before :

```

} catch (IOException e) {
    e.printStackTrace();
    System.err.println("Error saving user: " + e.getMessage());
    return false;
}

```

After Fixing :

```

} catch (IOException e) {
    LOGGER.log(Level.SEVERE, "Error saving user", e);
    return false;
}

```

18.

UserManagementController.java 1 day ago
UserManagementController.java 1 day ago
UserManagementController.java 1 day ago
UserManagementController.java 1 day ago
UserManagementController.java 1 day ago

- ✖ Lambda should be used to defer string concatenation.
- ✖ Lambda should be used to defer string concatenation.
- ✖ Lambda should be used to defer string concatenation.
- ✖ Lambda should be used to defer string concatenation.
- ✖ Lambda should be used to defer string concatenation.

Part of the code before fixing :

```

} catch (IOException e) {
    LOGGER.log(Level.SEVERE, "Error updating user: " + userId, e);
    return false;
}

```

After :

```
        }
    } catch (IOException e) {
        LOGGER.log(Level.SEVERE, e, () -> "Error updating user: " + userId);
        return false;
    }
}
```

19.

RegisterView.java	1 day ago	Remove the "sectorContainer" field and declare it as a local variable in the relevant methods.
RegisterView.java	1 day ago	Define a constant instead of duplicating this literal "Cashier" 4 times. [+4 locations]
RegisterView.java	1 day ago	Define a constant instead of duplicating this literal "Validation Error" 5 times. [+5 locations]
RegisterView.java	1 day ago	Define a constant instead of duplicating this literal "input-field" 3 times. [+3 locations]

Before :

```
Duplication
AlertDialog.showError( 1 "Validation_Error", "All fields are required.");
return false;
}
```

After :

```
private static final String USER_TYPE_ADMIN = "Administrator";
private static final String USER_TYPE_CASHIER = "Cashier";
private static final String USER_TYPE_MANAGER = "Manager";

private static final String STYLE_INPUT_FIELD = "input-field";
private static final String TITLE_VALIDATION_ERROR = "Validation_Error";
```

6. Actions and Decisions

During the code quality improvement phase, the project was analyzed using SonarQube to identify code issues and bad practices. Out of a total of 158 reported issues, 93 were successfully fixed. These fixes included reducing code duplication, lowering cognitive complexity, replacing deprecated methods, improving the use of Java Streams, and cleaning up unused or inefficient code. As a result, the codebase became cleaner, easier to understand, and more maintainable. This process helped improve the overall quality of the application and ensured better compliance with coding standards.