Code Coverage Analysis

# Instructions

1. Enter your team letter and name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Create screenshots of each tier-level page of the JaCoCo coverage report. Crop these screenshot as necessary so that the pertinent information is readable.
3. In your analysis, provide suggestions for improvements in the unit tests for each tier which are more insightful than the obvious "Write more tests" suggestion. There will be significant gaps in coverage due to the fact that you might not have unit tests for all of the project’s components. That’s expected.
4. In the last section of this file, identify two specific components, i.e. classes, that were unit tested. You will identify one with good code coverage and one with poor coverage.
5. Upload the final Word file to the *Code Coverage* dropbox in the **Team** **Exercises** category.

These statements in orange are additional instructions for the content for this analysis document. Perform the action then remove this text.

# Controller Tier

This is our analysis at the Controller Tier code for the project.

Attach a screenshot of the Controller Tier package JaCoCo report here.

## Analysis

Provide a short analysis of the Controller Tier package report here.

# Model Tier

This is our analysis at the Model Tier code for the project.

Attach a screenshot of the Model Tier package JaCoCo report here.

## Analysis

Provide a short analysis of the Model Tier package report here.

# Persistence Tier

This is our analysis at the Persistence Tier code for the project.

Attach a screenshot of the Persistence Tier package JaCoCo report here.

## Analysis

Provide a short analysis of the Persistence Tier package report here.

# Well-tested Component

This is our analysis of a well-tested component.

Attach a screenshot of the class-level (not the code-level) JaCoCo report here.

## Analysis

Provide a short analysis of what made the unit tests for this component so good.

# Poorly-tested Component

This is our analysis of a poorly-tested component.

Attach a screenshot of the class-level JaCoCo report here.

## Analysis

Provide an analysis of what the team will do to improve the quality (and coverage) of the unit tests for this component.