

## **Setup and General Notes**

**Faults and Weaknesses: (also available in the repository as a PDF)**

<https://docs.google.com/document/d/1PfFDYI560HOGMMn49ZT60TdjnTiGGoqIGYGq5m3I5h4/edit?usp=sharing>

**Unachievable Coverage: (also available in the repository as a PDF)**

<https://docs.google.com/document/d/1gvAH4OxvRCTo1MY0o9kkozNLS-2ueFoUFzNM8MOPcmM/edit>

**Slides:**

[https://docs.google.com/presentation/d/1N\\_DPQQu7EafgJz-lSWLJX1dcG6w1JwDfLbgIfTvV94/edit?usp=sharing](https://docs.google.com/presentation/d/1N_DPQQu7EafgJz-lSWLJX1dcG6w1JwDfLbgIfTvV94/edit?usp=sharing)

**The submission should contain:**

- This file saved as a PDF, located within the GitHub
- The above two files saved as PDF's, located within the GitHub
- A directory containing our completed version of the project with no changes to the source code aside from adding tests, which is a private version of the original repository:

<https://github.com/shashirajraja/shopping-cart/>


- Contains a coverage report generated
  - Slides can be found at the above Google Slides link
- 1) To set up, follow the README within the GitHub repo.
    - a) Don't forget to change your application properties to match MySQL and Gmail credentials! Don't forget to disable MySQL safe mode!
  - 2) How to reproduce results

- a) Tests are located within /src/tests, with three categories of tests available
- b) Tests generally assume the default setup was performed prior to running them, and restore the database to its previous condition whenever possible. If a test fails with “failed setup,” then this may not have been the case.

**c) Faults**

- i) Tests that uncover faults or fail in general will have often have a comment indicating why they fail, but if not, please refer to the Faults and Weaknesses file
- ii) Additional details for faults can be found in the “Faults and Weaknesses” document located in the submission folder.

**d) Whitebox / Blackbox / Coverage**

- i) Whitebox, blackbox tests are available under src/test/ImplTesting
- ii) After proper setup, they can run as regular JUnit tests, with results reported as expected.
- iii) By creating a new JUnit configuration in the top right in IntelliJ for the “test” package, selecting the option to enable tracing for branch coverage, and the option to choose an alternative coverage runner (Jacoco)
- iv) Clicking the configuration by the arrow will introduce a drop-down that will allow you to run the configuration with coverage. Then, when coverage is generated in the right-side panel, you can click  in order to generate a coverage report. For your convenience, a pre-generated coverage report will already be available under the “htmlReport” directory within the submission itself.

- v) Additional details for missed coverage can be found in the “Unachievable Coverage” document located in the submission folder.
- vi) Some test files have a significantly higher number of failures: this is due to exploring two separate testing philosophies. Out of the six files, half assume intended behavior (i.e. throwing an Exception on a null input is necessary), while the other half assumes that an error state returned (i.e. returning a false boolean) is sufficient.

**e) GUI Tests**

- i) Available under src/test/GUITesting
- ii) Begin the Tomcat server, and then you can run the tests

**f) API Tests**

- i) Available under src/test/APITesting
- ii) Begin the Tomcat server, and then you can run the tests

**g) Usability Study Results**

- i) Located in the root of the project as “User Feedback.docx”