Jesse Arstein

Jay Van Alstyne

HW 4

Question 1:

a)

1. We downloaded the Zerocode system.
2. Zerocode is a framework that makes it easier to write automated tests against API interfaces. It basically makes API calls into easy to call JSON files and makes it easy to do security, stress and load testing.
3. There is 34296 lines of code in the GitHub repository. We calculated this by using a chrome extension that counts the lines of code within a GitHub repository.

b)

1. Found 22 files that possibly contain design patterns.

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ExtentReportsFactory.java

Possible patterns: Factory

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\HighChartColumnHtmlBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeChartKeyValueArrayBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeChartKeyValueBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeCsvReportBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeExecReportBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeIoWriteBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeReportStepBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\engine\preprocessor\ScenarioExecutionState.java

Possible patterns: State

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\engine\preprocessor\StepExecutionState.java

Possible patterns: State

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\httpclient\ssl\CorporateProxyNoSslContextHttpClient.java

Possible patterns: Proxy

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\httpclient\ssl\SslTrustCorporateProxyHttpClient.java

Possible patterns: Proxy

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\logbuilder\RequestLogBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\logbuilder\ResponseLogBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\main\java\org\jsmart\zerocode\core\logbuilder\ScenarioLogBuilder.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\domain\builders\ZeroCodeIoWriteBuilderTest.java

Possible patterns: Builder

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\engine\preprocessor\ScenarioExecutionStateTest.java

Possible patterns: State

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\engine\preprocessor\StepExecutionStateTest.java

Possible patterns: State

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\httpclient\ssl\SslTrustCorporateProxyHttpClientTest.java

Possible patterns: Proxy

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\runner\retry\RetryWithStateTest.java

Possible patterns: State

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\soap\SoapCorpProxySslHttpClientTest.java

Possible patterns: Proxy

D:\College Files\Fall 2019\Software Engineering\HW4\zerocode-master\core\src\test\java\org\jsmart\zerocode\core\tests\customrunner\SimulatorState.java

Possible patterns: State

2. The design pattern finder tool appears to have an xml file with a list of pattern names. The tool then searches through what you specify for files and sees if they contains a name of a pattern mentioned in any of the files. Its really just a simple string comparison.

3. We don’t believe the process the tool uses is correct, for example in java there is the string builder library that can be utilized in your code. Since that string would show up in the search it might flag our file as using the builder design pattern which it might not be using. If we were to design a program that finds design patterns it might be a good plan to start and see if project files contain those design pattern keywords. We could also monitor the memory addresses that the program accesses and how the memory is changed by the program. Following those memory patterns might indicate a design pattern. Realistically, it would probably have to be a combination of a few different ways analyzing the code.

Question 2:

1. We used the web interface to upload the files for the most part.
2. We used git clone, echo “test file” >> test.txt, git add, git commit -m and git push. Attached is a screenshot of each of those outputs and a screenshot of the github

Link to github : <https://github.com/jessejohn01/ESOF322> (Can’t click on paper but..)

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated