CodingInfinity

Benchmark Service Testing Documentation

Git:
`https://github.com/CodingInfinity/Benchmark-Service-Documentation`
GitHub Organisation: `https://github.com/CodingInfinity`

The Client:
Ms Vreda Pieterse
Department of Computer Science
University Of Pretoria

The Team:
Andrew Broekman *11089777*
Brenton Watt *14032644*
Fabio Loreggian *14040426*
Reinhardt Cromhout *14009936*

September 2016

Contents

1 Introduction

This is the testing documentation for the Benchmark Service. It details how each component of the system is tested.

1.1 Purpose

This document combines the unit test plan and report into a single coherent artifact. The overall broad purpose of this system is to allow a user to upload an algorithm and have it benchmarked with a dataset, the algorithm will be benchmarked according to wall-clock time, CPU usage and memory usage. This document serves to illustrate the tests we have implemented to ensure the system is working correctly. Test driven development is essential to our project as it allows us to know what areas of our project are working as they should and which areas are either causing errors or not returning the expected results.

1.2 Scope

The scope of this document is structured as follows. The features that are considered for testing are listed in section .... Tests that have been identified from the requirements are discussed in detail in section .... Furthermore, this document outlines the test environment and the risks involved in the testing approaches that will be followed. Assumptions and dependencies of this test plan will also be mentioned. Section 5.1 and 6.2 outlines, discusses and concludes on the results of the tests, respectively.

1.3 Test Environment

1. Programming Languages:
    - Java
2. Testing Frameworks:
    - JUnit
    - Mockito
    - Powermock with Spring
3. Operating System
    - Linux
4. Web Browsers
    - Google Chrome

1.4 Assumptions and Dependencies

To run the tests, the following is assumed to be in place:

- Cloned version of the Benchmark Management Repository
- Java 7+ JVM
- Maven 3.0.0+

There are no additional dependecies required, as Maven will download and setup required dependencies to perform testing.

2 Unit Test Plan

2.1 Benchmark Management Server

The Benchmark Management Server is a Maven based project, and as such makes use of the Maven framework to assist in executing tests. All tests for the system can be located in the `com.codinginfinity.benchmark.management.test` package located in the `/src/test/java/` folder.

As such, only some tests will be illustrated. The reader is however referred to the source code for the remainder of the tests.

2.1.1 Functional Feature to be Tested

All functional use cases, as outlined in the document `Benchmark Service Functional Requirements Documentation`, have associated test implemented in the system. The stated document also specifies required constraints, pre- and post-conditions as well as any other specific detail to be tested within the relevant diagrams.

Unit tests for the use cases will be implemented as black and white box tests.

3 Test Cases

As all use cases in the system have implemented unit tests, the following specifications will be propegated down to every unit test.

Tests cases for the following modules exists:

- Experiment Management
- Notifications
- Reporting
- Repository Management
- User Management

3.1 Objetive

The purpose of the unit test is to ensure that:

- Pre-conditions are checked and raise appropriate exceptions when not met
- Post conditions are fulfilled
- Stated functional requirements specific to the use is implemented

3.2 Input

For every use case, both valid and invalid entries will be construcuted to ensure that exceptions are raised if pre-condiions are not met. Further all return values are checked to ensure that the returned object is correct according the stated functional requirements.

3.3 Outcome

Every use case should fulfill its stated service contract by returning the appropriate respone object as outlined in the the functional requirements document.

4 Pass/Fail Criteria

For every use case tested, the following criteria must be met for the test to be considered a **Pass**.

- All pre-conditions should be fulfilled. If not, an appropriate execption must be raised as to alert the client.
- All post-condtions must hold upon returning from the function call. If a post condition doesn't hold, the test will fail.
- Any stated functionality specific to the use case as set out in the functional requirements document must be validated within the unit test to ensure stated functionality is implemented.

5 Unit Test Report

5.1 Detailed Test Results

5.1.1 Overview of test results

All 203 of the implemented unit tests in the Benchmark Management System passed successfully as can be observed on `https://travis-ci.org/CodingInfinity/Benchmark-Management-Serve`

Refer to section 1.3 for a list of the frameworks used to test the management system. These frameworks where selected as they are currently the most widely used frameworks in Java, and because most open source projects utlize these frameworks. It is important

to select frameworks that are recoginzed by the open source community, as the project is envisage to become an open source project.

5.1.2 Functional requirements test results

Refer to section 2.1 on where the tests for the project in question can be located.

Table 1: Overview of unit tests in packages located in `com.codinginfinity.benchmark.management`

Package	Number of Unit Tests	Number of Test Passed
test	2	2
test.domain	44	44
test.security	21	21
test.service	136	136

Table 2: Overview of unit tests in the Experiment Management service contract

Use Case	Number of Unit Tests	Number of Test Passed
Create Experiment	0	0
Get All User Experiments	2	2
Get All User Experiments	2	2
Get Experiment By ID	2	2
Get Experiment Weekly Report	0	0
Get Node Status By ID	2	2
Is Job On Queue	4	4
Register Node Heartbeat	1	1
Remove Node	2	2
Save Job Results	0	0

Table 3: Overview of unit tests in the Notification service contract

Use Case	Number of Unit Tests	Number of Test Passed
Send Activation Email	1	1
Send Creation Email	1	1
Send Email	2	2
Send Password Reset Email	1	1

Table 4: Overview of unit tests in the Reporting service contract

Use Case	Number of Unit Tests	Number of Test Passed
Download Results	2	2

Table 5: Overview of unit tests in the Repository Management service contract for Dataset Management and Retrieval

Use Case	Number of Unit Tests	Number of Test Passed
Add Dataset	6	6
Delete Dataset	2	2
Get All Datasets	2	2
Get Dataset By Category	2	2
Get Dataset By ID	2	2
Get Dataset By Username	2	2
Update Dataset Metadata	2	2

Table 6: Overview of unit tests in the Repository Management service contract for Dataset Category Management and Retrieval

Use Case	Number of Unit Tests	Number of Test Passed
Add Dataset Category	2	2
Delete Dataset Category	2	2
Get All Dataset Categories	2	2
Get Dataset Category By ID	2	2
Get Dataset Category By Name	2	2
Update Dataset Category	2	2

Note that the unit tests for the Algorithm and Algorithm Category Management and Retrieval is mutatis mutandis the same as for the Dataset and Datatset Category Management and Retrieval.

Table 7: Overview of unit tests in the User Management service contract

Use Case	Number of Unit Tests	Number of Test Passed
Activate Registration	2	2
Change Password	2	2
Compelete Password Reset	3	3
Create Managed User	5	5
Create Unmanaged User	3	3
Delete User	2	2
Get User With Authorities By Login	2	2
Request Password Reset	4	4
Update User	3	3

6 Other

6.1 POM file for Benchmark Management Server

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <name>Benchmark Management Server</name>
  <description>Management backend and co-ordination platform for the benchmark service platform</description>
  <inceptionYear>2016</inceptionYear>

  <prerequisites>
    <maven>3.0.0</maven>
  </prerequisites>

  <parent>
    <artifactId>spring-boot-starter-parent</artifactId>
    <groupId>org.springframework.boot</groupId>
    <version>1.3.5.RELEASE</version>
    <relativePath />
  </parent>

  <groupId>com.codinginfinity.benchmark</groupId>
  <artifactId>management</artifactId>
  <version>0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <organization>
    <name>Department of Computer Science, University of Pretoria</name>
    <url>http://www.cs.up.ac.za</url>
  </organization>

  <licenses>
    <license>
      <name>GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007</name>
      <url>https://www.gnu.org/licenses/agpl-3.0.txt</url>
    </license>
  </licenses>

  <scm>
    <url>scm:git:git@github.com:CodingInfinity/Benchmark-Management-Server.git</url>
    <connection>scm:git:git@github.com:CodingInfinity/Benchmark-Management-Server.git</connection>
    <developerConnection>scm:git:git@github.com:CodingInfinity/Benchmark-Management-Server.git</developerConnection>
  </scm>

```

```
43  <developers>
44    <developer>
45      <id>AndrewBroekman@tuks.co.za</id>
46      <name>Andrew Broekman</name>
47      <organization>University of Pretoria</organization>
48      <email>AndrewBroekman@tuks.co.za</email>
49    </developer>
50
51    <developer>
52      <id>u14040426@tuks.co.za</id>
53      <name>Fabio Loreggian</name>
54      <organization>University of Pretoria</organization>
55      <email>u14040426@tuks.co.za</email>
56    </developer>
57
58    <developer>
59      <id>BrentonWatt@tuks.co.za</id>
60      <name>Brenton Watt</name>
61      <organization>University of Pretoria</organization>
62      <email>BrentonWatt@tuks.co.za</email>
63    </developer>
64
65    <developer>
66      <id>u14009936@tuks.co.za</id>
67      <name>Reinhardt Cromhout</name>
68      <organization>University of Pretoria</organization>
69      <email>u14009936@tuks.co.za</email>
70    </developer>
71  </developers>
72
73  <ciManagement>
74    <system>Travis CI</system>
75    <url>https://travis-ci.org/CodingInfinity/Benchmark-Management-Server</
      url>
76  </ciManagement>
77
78  <issueManagement>
79    <system>Github Issues</system>
80    <url>https://github.com/CodingInfinity/Benchmark-Management-Server/
      issues</url>
81  </issueManagement>
82
83  <properties>
84    <activemq.version>5.13.3</activemq.version>
85    <camel.version>2.17.1</camel.version>
86    <commons-compress.version>1.11</commons-compress.version>
87    <commons-lang.version>2.6</commons-lang.version>
88    <commons-io.version>2.5</commons-io.version>
89    <dbh2.version>1.4.192</dbh2.version>
90    <hibernate.version>4.3.11.Final</hibernate.version>
91    <jacoco-maven-plugin.version>0.7.6.201602180812</jacoco-maven-plugin.
      version>
92    <java.version>1.8</java.version>
```

```
93      <javax.inject.version>1</javax.inject.version>
94      <javax.validation.version>1.1.0.Final</javax.validation.version>
95      <junit.version>4.12</junit.version>
96      <liquibase-hibernate4.version>3.5</liquibase-hibernate4.version>
97      <liquibase-slf4j.version>1.2.1</liquibase-slf4j.version>
98      <liquibase.version>3.4.2</liquibase.version>
99      <lombok.version>1.16.8</lombok.version>
100     <maven-enforcer-plugin.version>1.4.1</maven-enforcer-plugin.version>
101     <maven.compiler.source>${java.version}</maven.compiler.source>
102     <maven.compiler.target>${java.version}</maven.compiler.target>
103     <mockito.version>2.0.2-beta</mockito.version>
104     <powermock.version>1.6.5</powermock.version>
105     <project.testresult.directory>${project.build.directory}/test-results</
        project.testresult.directory>
106     <sortpom-maven-plugin.version>2.5.0</sortpom-maven-plugin.version>
107     <spring-framework.version>4.2.0.RELEASE</spring-framework.version>
108     <spring-security-oauth2.version>2.0.10.RELEASE</spring-security-oauth2.
        version>
109     <springfox.version>2.5.0</springfox.version>
110     <thrift.version>0.9.3</thrift.version>
111   </properties>
112
113   <dependencies>
114
115     <!-- ActiveMQ Dependencies -->
116     <dependency>
117       <groupId>org.apache.activemq</groupId>
118       <artifactId>activemq-broker</artifactId>
119       <version>${activemq.version}</version>
120     </dependency>
121     <dependency>
122       <groupId>org.apache.activemq</groupId>
123       <artifactId>activemq-camel</artifactId>
124       <version>${activemq.version}</version>
125     </dependency>
126
127     <!-- Apache Camel -->
128     <dependency>
129       <groupId>org.apache.camel</groupId>
130       <artifactId>camel-amqp</artifactId>
131       <version>${camel.version}</version>
132     </dependency>
133     <dependency>
134       <groupId>org.apache.camel</groupId>
135       <artifactId>camel-core</artifactId>
136       <version>${camel.version}</version>
137     </dependency>
138     <dependency>
139       <groupId>org.apache.camel</groupId>
140       <artifactId>camel-spring-boot-starter</artifactId>
141       <version>${camel.version}</version>
142     </dependency>
143
```

```
    <!-- Apache Compress -->
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-compress</artifactId>
      <version>${commons-compress.version}</version>
    </dependency>

    <!-- Apache Thrift -->
    <dependency>
      <groupId>org.apache.thrift</groupId>
      <artifactId>libthrift</artifactId>
      <version>${thrift.version}</version>
    </dependency>

    <!-- Apache Commons Dependencies -->
    <dependency>
      <groupId>commons-lang</groupId>
      <artifactId>commons-lang</artifactId>
      <version>${commons-lang.version}</version>
    </dependency>
    <dependency>
      <groupId>commons-io</groupId>
      <artifactId>commons-io</artifactId>
      <version>${commons-io.version}</version>
    </dependency>

    <dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-swagger-ui</artifactId>
      <version>${springfox.version}</version>
    </dependency>

    <!-- H2 Database -->
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <version>${dbh2.version}</version>
    </dependency>

    <dependency>
      <groupId>com.fasterxml.jackson.dataformat</groupId>
      <artifactId>jackson-dataformat-csv</artifactId>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.datatype</groupId>
      <artifactId>jackson-datatype-jsr310</artifactId>
    </dependency>

    <!-- Java EE Dependencies -->
    <dependency>
      <groupId>javax.inject</groupId>
      <artifactId>javax.inject</artifactId>
      <version>${javax.inject.version}</version>
```

```
197     </dependency>
198     <dependency>
199       <groupId>javax.validation</groupId>
200       <artifactId>validation-api</artifactId>
201       <version>${javax.validation.version}</version>
202     </dependency>
203
204     <!-- Liquibase -->
205     <dependency>
206       <groupId>com.mattbertolini</groupId>
207       <artifactId>liquibase-slf4j</artifactId>
208       <version>${liquibase-slf4j.version}</version>
209     </dependency>
210     <dependency>
211       <groupId>org.liquibase</groupId>
212       <artifactId>liquibase-core</artifactId>
213       <exclusions>
214         <exclusion>
215         <artifactId>jetty-servlet</artifactId>
216           <groupId>org.eclipse.jetty</groupId>
217         </exclusion>
218       </exclusions>
219     </dependency>
220
221     <!-- Lombok -->
222     <dependency>
223       <groupId>org.projectlombok</groupId>
224       <artifactId>lombok</artifactId>
225       <version>${lombok.version}</version>
226       <scope>provided</scope>
227     </dependency>
228
229     <!-- Springframework Dependencies -->
230     <dependency>
231       <groupId>org.springframework</groupId>
232       <artifactId>spring-jms</artifactId>
233     </dependency>
234     <dependency>
235       <groupId>org.springframework.security.oauth</groupId>
236       <artifactId>spring-security-oauth2</artifactId>
237       <version>${spring-security-oauth2.version}</version>
238     </dependency>
239
240     <!-- Spring Boot Dependencies -->
241     <dependency>
242       <groupId>org.springframework.boot</groupId>
243       <artifactId>spring-boot-autoconfigure</artifactId>
244     </dependency>
245     <dependency>
246       <groupId>org.springframework.boot</groupId>
247       <artifactId>spring-boot-configuration-processor</artifactId>
248       <optional>true</optional>
249     </dependency>
```

```
250     <dependency>
251       <groupId>org.springframework.boot</groupId>
252       <artifactId>spring-boot-starter-aop</artifactId>
253     </dependency>
254     <dependency>
255       <groupId>org.springframework.boot</groupId>
256       <artifactId>spring-boot-starter-data-jpa</artifactId>
257     </dependency>
258     <dependency>
259       <groupId>org.springframework.boot</groupId>
260       <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
261     </dependency>
262     <dependency>
263       <groupId>org.springframework.boot</groupId>
264       <artifactId>spring-boot-starter-mail</artifactId>
265     </dependency>
266     <dependency>
267       <groupId>org.springframework.boot</groupId>
268       <artifactId>spring-boot-starter-thymeleaf</artifactId>
269     </dependency>
270     <dependency>
271       <groupId>org.springframework.boot</groupId>
272       <artifactId>spring-boot-starter-web</artifactId>
273     </dependency>
274
275     <!-- Swagger Dependencies -->
276     <dependency>
277       <groupId>io.springfox</groupId>
278       <artifactId>springfox-swagger2</artifactId>
279       <version>${springfox.version}</version>
280       <exclusions>
281         <exclusion>
282         <artifactId>mapstruct</artifactId>
283           <groupId>org.mapstruct</groupId>
284         </exclusion>
285       </exclusions>
286     </dependency>
287
288
289     <!-- Testing Framework Dependencies -->
290     <dependency>
291       <groupId>junit</groupId>
292       <artifactId>junit</artifactId>
293       <version>${junit.version}</version>
294       <scope>test</scope>
295     </dependency>
296     <dependency>
297       <groupId>net.trajano.commons</groupId>
298       <artifactId>commons-testing</artifactId>
299       <version>1.0.1</version>
300       <scope>test</scope>
301     </dependency>
302     <dependency>
```

```
303         <groupId>org.hsqldb</groupId>
304         <artifactId>hsqldb</artifactId>
305         <scope>test</scope>
306       </dependency>
307       <dependency>
308         <groupId>org.mockito</groupId>
309         <artifactId>mockito-all</artifactId>
310         <version>${mockito.version}</version>
311         <scope>test</scope>
312       </dependency>
313       <dependency>
314         <groupId>org.powermock</groupId>
315         <artifactId>powermock-api-mockito</artifactId>
316         <version>${powermock.version}</version>
317         <scope>test</scope>
318       </dependency>
319       <dependency>
320         <groupId>org.powermock</groupId>
321         <artifactId>powermock-core</artifactId>
322         <version>${powermock.version}</version>
323         <scope>test</scope>
324       </dependency>
325       <dependency>
326         <groupId>org.powermock</groupId>
327         <artifactId>powermock-module-junit4</artifactId>
328         <version>${powermock.version}</version>
329         <scope>test</scope>
330       </dependency>
331       <dependency>
332         <groupId>org.springframework</groupId>
333         <artifactId>spring-test</artifactId>
334         <version>${spring-framework.version}</version>
335         <scope>test</scope>
336       </dependency>
337       <dependency>
338         <groupId>org.subethamail</groupId>
339         <artifactId>subethasmtp</artifactId>
340         <version>3.1.7</version>
341         <scope>test</scope>
342       </dependency>
343     </dependencies>
344
345     <build>
346       <defaultGoal>spring-boot:run</defaultGoal>
347       <plugins>
348         <plugin>
349           <groupId>com.github.ekryd.sortpom</groupId>
350           <artifactId>sortpom-maven-plugin</artifactId>
351           <version>${sortpom-maven-plugin.version}</version>
352           <executions>
353             <execution>
354               <phase>verify</phase>
355               <goals>
```

```
356                 <goal>sort</goal>
357               </goals>
358             </execution>
359           </executions>
360           <configuration>
361             <sortProperties>true</sortProperties>
362             <nrOfIndentSpace>4</nrOfIndentSpace>
363             <sortDependencies>groupId,artifactId</sortDependencies>
364             <sortPlugins>groupId,artifactId</sortPlugins>
365             <keepBlankLines>true</keepBlankLines>
366             <expandEmptyElements>false</expandEmptyElements>
367           </configuration>
368         </plugin>
369         <plugin>
370           <groupId>org.apache.maven.plugins</groupId>
371           <artifactId>maven-eclipse-plugin</artifactId>
372           <configuration>
373             <downloadSources>true</downloadSources>
374             <downloadJavadocs>true</downloadJavadocs>
375           </configuration>
376         </plugin>
377         <plugin>
378           <groupId>org.apache.maven.plugins</groupId>
379           <artifactId>maven-enforcer-plugin</artifactId>
380           <version>${maven-enforcer-plugin.version}</version>
381           <executions>
382             <execution>
383               <id>enforce-versions</id>
384               <goals>
385                 <goal>enforce</goal>
386               </goals>
387             </execution>
388           </executions>
389           <configuration>
390             <rules>
391               <requireMavenVersion>
392                 <message>You are running an older version of Maven. Benchmark
    Management Server requires at least Maven 3.0</message>
393                 <version>[3.0.0,)</version>
394               </requireMavenVersion>
395               <requireJavaVersion>
396                 <message>You are running an older version of Java.  Benchmark
    Management Server requires at least JDK ${java.version}</message>
397                 <version>[${java.version}.0,)</version>
398               </requireJavaVersion>
399             </rules>
400           </configuration>
401         </plugin>
402         <plugin>
403           <groupId>org.apache.maven.plugins</groupId>
404           <artifactId>maven-surefire-plugin</artifactId>
405           <configuration>
```

```
406             <argLine>-Djava.security.egd=file:/dev/./urandom -Xmx256m ${
    surefireArgLine}</argLine>
407             <runOrder>alphabetical</runOrder>
408           </configuration>
409         </plugin>
410         <plugin>
411           <groupId>org.eluder.coveralls</groupId>
412           <artifactId>coveralls-maven-plugin</artifactId>
413           <version>4.2.0</version>
414         </plugin>
415         <plugin>
416           <groupId>org.jacoco</groupId>
417           <artifactId>jacoco-maven-plugin</artifactId>
418           <version>${jacoco-maven-plugin.version}</version>
419           <executions>
420             <execution>
421               <id>pre-unit-test</id>
422               <goals>
423                 <goal>prepare-agent</goal>
424               </goals>
425               <configuration>
426                 <propertyName>surefireArgLine</propertyName>
427               </configuration>
428             </execution>
429             <execution>
430               <id>post-unit-test</id>
431               <phase>test</phase>
432               <goals>
433                 <goal>report</goal>
434               </goals>
435             </execution>
436             <execution>
437               <id>default-report</id>
438               <phase>prepare-package</phase>
439               <goals>
440                 <goal>report</goal>
441               </goals>
442             </execution>
443           </executions>
444           <configuration>
445             <excludes>
446               <exclude>**/**Exception.class</exclude>
447               <exclude>**/aop/**/*.class</exclude>
448               <exclude>**/config/**/*.class</exclude>
449               <exclude>**/domain/**/*.class</exclude>
450               <exclude>**/dto/**/*.class</exclude>
451               <exclude>**/http/**/*.class</exclude>
452               <exclude>**/jackson/mixin/**/*.class</exclude>
453               <exclude>**/service/**/exception/*.class</exclude>
454               <exclude>**/service/**/response/*.class</exclude>
455               <exclude>**/service/**/request/*.class</exclude>
456               <exclude>**/thrift/**/*.class</exclude>
457               <exclude>**/web/rest/**/*.class</exclude>
```

```
458             </excludes>
459           </configuration>
460         </plugin>
461         <plugin>
462           <groupId>org.springframework.boot</groupId>
463           <artifactId>spring-boot-maven-plugin</artifactId>
464           <configuration>
465             <executable>true</executable>
466             <arguments>
467               <argument>--spring.profiles.active=dev</argument>
468             </arguments>
469           </configuration>
470         </plugin>
471         <plugin>
472           <groupId>org.liquibase</groupId>
473           <artifactId>liquibase-maven-plugin</artifactId>
474           <version>${liquibase.version}</version>
475           <configuration>
476             <changeLogFile>src/main/resources/config/liquibase/master.xml</
    changeLogFile>
477             <diffChangeLogFile>src/main/resources/config/liquibase/changelog
    /${maven.build.timestamp}_changelog.xml</diffChangeLogFile>
478             <driver>org.postgresql.Driver</driver>
479             <url>jdbc:postgresql://localhost:5432/postgresql</url>
480             <defaultSchemaName></defaultSchemaName>
481             <username>postgresql</username>
482             <password></password>
483             <referenceUrl>hibernate:spring:com.mycompany.myapp.domain?dialect
    =org.hibernate.dialect.PostgreSQL82Dialect&amp;hibernate.ejb.
    naming_strategy=org.springframework.boot.orm.jpa.hibernate.
    SpringNamingStrategy</referenceUrl>
484             <verbose>true</verbose>
485             <logging>debug</logging>
486             <diffExcludeObjects>oauth_access_token, oauth_approvals,
    oauth_client_details, oauth_client_token, oauth_code,
    oauth_refresh_token</diffExcludeObjects>
487           </configuration>
488           <dependencies>
489             <dependency>
490               <groupId>org.javassist</groupId>
491               <artifactId>javassist</artifactId>
492               <version>3.18.2-GA</version>
493             </dependency>
494             <dependency>
495               <groupId>org.liquibase.ext</groupId>
496               <artifactId>liquibase-hibernate4</artifactId>
497               <version>${liquibase-hibernate4.version}</version>
498             </dependency>
499             <dependency>
500               <groupId>org.springframework.boot</groupId>
501               <artifactId>spring-boot-starter-data-jpa</artifactId>
502               <version>${project.parent.version}</version>
503             </dependency>
```

```
        </dependencies>
      </plugin>
      <plugin>
        <groupId>com.spotify</groupId>
        <artifactId>docker-maven-plugin</artifactId>
        <version>0.4.5</version>
        <configuration>
          <imageName>benchmark</imageName>
          <dockerDirectory>src/main/docker</dockerDirectory>
          <resources>
            <resource>
              <targetPath>/</targetPath>
              <directory>${project.build.directory}</directory>
              <include>${project.build.finalName}.war</include>
            </resource>
          </resources>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-report-plugin</artifactId>
        <version>2.19</version>
      </plugin>
    </plugins>
  </reporting>

  <profiles>
    <profile>
      <id>dev</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <logback.loglevel>DEBUG</logback.loglevel>
      </properties>
    </profile>

    <profile>
      <id>prod</id>
      <build>
        <plugins>
          <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
              <executable>true</executable>
              <arguments>
                <argument>--spring.profiles.active=prod</argument>
```

```
557                 </arguments>
558               </configuration>
559             </plugin>
560           </plugins>
561         </build>
562         <properties>
563           <logback.loglevel>INFO</logback.loglevel>
564         </properties>
565       </profile>
566     </profiles>
567 </project>
```

6.2 Conclusion

The Benchmark Instrumentation Application and Benchmark Web Application was not tested as no mature and standard way exsists to implement unit testing within the said frameworks or languages. The difficulty encountered in the testing a Web Application is that the application only issues RESTful calls to the backend. As such no logic remain on the client side to be tested, which improves software reliability and security.

Further the Instrumentation software is built as a monolithic application, with developer optimized code and routines as to elimate any possible side effects which application may introduce. This optimization by the developers make software very hard to test, as no component can really be tested in isolation.