Code Inspection

05/01/2016

Elis Bardhi **790135**

Andrea Cavalli **841512**

Mario Dolci **773705**

1. Class description

1.1. Assigned class

Class path: appserver/ejb/ejb-container/src/main /java/com/sun/ejb/containers/
EJBTimerService.java

1.2. Function of the class

EJBTimerService is the central controller of the EJB timer service. There is one instance of EJBTimerService per VM. All operations and state transitions on timers pass through this class in order to reduce the overall complexity by encapsulating the timer logic within this class and by keeping the supporting classes simple.

1.3. Functions

- **F1:** **private void** deliverTimeout(TimerPrimaryKey timerId) line 1105
- **F2:** **boolean** postEjbTimeout(TimerPrimaryKey timerId) line 1321
- **F3:** **private void** cancelTimerSynchronization(EJBContextImpl context_, TimerPrimaryKey timerId, **long** containerId, String ownerId, **boolean** persistent) line 1519
- **F4:** ContainerSynchronization getContainerSynch(EJBContextImpl context_,String timerId, **boolean** persistent) line 1588

2. List of issues found by applying the checklist

2.1. Naming Conventions

All class names, interface names, method names, class variables, method variables, and constants used should have meaningful names and do what the name suggests.

Class:	**protected boolean** isDas – The meaning of the variable was not found.
F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

If one-character variables are used, they are used only for temporary "throwaway" variables, such as those used in for loops.

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	In line 1597 there is a one-character variable used to have a more readable code: ComponentInvocation i = ejbContainerUtil.getCurrentInvocation().

Class names are nouns, in mixed case, with the first letter of each word in capitalized.

Class: No issues.

Interface names should be capitalized like classes.

Class: No issues.

Method names should be verbs, with the first letter of each addition word capitalized.

Class: No issues.

Class variables, also called attributes, are mixed case, but might begin with an underscore ('_') followed by a lowercase first letter. All the remaining words in the variable name have their first letter capitalized.

Class: Attributes in lines 91,93,94,95,97,107,110,112,121,137,141,144,149,153,158,160 does not start with "_" character.

Constants are declared using all uppercase with words separated by an underscore.

Class:	No issues.

2.2. Indention

Three or four spaces are used for indentation and done so consistently

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

No tabs are used to indent

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

2.3. Braces

Consistent bracing style is used, either the preferred "Allman" style or the "Kernighan and Ritchie" style.

F1:	It is used only "Kernighan and Ritchie" style;
F2:	It is used only "Kernighan and Ritchie" style;
F3:	It is used only "Kernighan and Ritchie" style;
F4:	It is used only "Kernighan and Ritchie" style.

All if, while, do-while, try-catch, and for statements that have only one statement to execute are surrounded by curly braces.

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

2.4. File Organization

Blank lines and optional comments are used to separate sections (beginning comments, package/import statements, class/interface declarations which include class variable/attributes declarations, constructors, and methods).

Class:	This is not respected on the import section.

Where practical, line length does not exceed 80 characters.

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	Line 1600 is 89 characters long, line 1605 is 98 characters long.

When line length must exceed 80 characters, it does NOT exceed 120 characters.

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

2.5. Wrapping Lines

Line break occurs after a comma or an operator.

F1:	No issues;
F2:	Lines 1359 and 1372 break after a '+' operator;
F3:	Lines 1519, 1520, 1556 and 1557 break after a comma;
F4:	The line 1588 breaks after a comma and lines 1605, 1614 break after a '+' operator.

Higher-level breaks are used.

F1:	No issues;
F2:	No issues;
F3:	No issues;
F4:	No issues.

A new statement is aligned with the beginning of the expression at the same level as the previous line.

F1:	No issues;
F2:	No issues
F3:	No issues;
F4:	No issues.

2.6. Comments

Comments are used to adequately explain what the class, interface, methods, and blocks of code are doing.

Class:	Not all the functions are commented;
F1:	The method and its blocks of code are commented;
F2:	The method and its blocks of code are commented;
F3:	The method and its blocks of code are commented except some getters;
F4:	The method and its blocks of code are commented except some getters.

Commented out code contains a reason for being commented out and a date it can be removed from the source file if determined it is no longer needed.

Class:	The class contains two commented out statements in line 1606 and 1901

2.7. Java Source Files

Each Java source file contains a single public class or interface.

Our java source file (EJBTimerService.java) contain two public classes:

- **public class** EJBTimerService
- **public static class** TimerCache

The public class is the first class or interface in the file.

Correct.

Check that the external program interfaces are implemented consistently with what is described in the javadoc.

Class does not have an interface.

Check that the javadoc is complete

Javadoc is not complete. It does not contain:

- **private void** deliverTimeout(TimerPrimaryKey timerId)

2.8. Package and Import Statements

If any package statements are needed, they should be the first non-comment statements. Import statements follow.

Correct.

2.9. Class and Interface Declarations

The class or interface declarations shall be in the following order:

A. class/interface documentation comment

B. class or interface statement

C. class/interface implementation comment, if necessary

D. class (static) variables

a. first public class variables

b. next protected class variables

c. next package level (no access modifier)

d. last private class variables

This section is not respected. This is the current order of declarations that is wrong:

```
...
private long nextTimerIdMillis_ = 0;
private long nextTimerIdCounter_ = 0;
private String domainName_;
protected boolean isDas;
public static final int STATE_ACTIVE    = 0;
public static final int STATE_CANCELLED = 1;
...
```

E. instance variables

a. first public instance variables

e. next protected instance variables

f. next package level (no access modifier)

g. last private instance variables

Order respected.

F. constructors

G. methods

The class interface respects this order.

Methods are grouped by functionality rather than by scope or accessibility.

We noted that methods are grouped by functionality. We can note this by the name of the methods.

Check that the code is free of duplicates, long methods, big classes, breaking encapsulation, as well as if coupling and cohesion are adequate.

By using SonarQube we notice that our class EJBTimerService has a complexity of 310, which is greater than 200 authorized.

F1: has a Cyclomatic Complexity of 32 which is greater than 10, that is the maximum authorized value for a function.

2.10. Initialization and Declarations

Check that variables and class members are of the correct type. Check that they have the right visibility (public/private/protected)

No issues.

Check that variables are declared in the proper scope.

No issues.

Check that constructors are called when a new object is desired

Not always, for example inside F1 in the lines 1122 or 1136 the constructor is not called.
Inside F2, in the lines 1334 and 1340 the constructor is not called.
Inside F3, in line 1533 the constructor is not called.

Check that all object references are initialized before use

No issues.

Variables are initialized where they are declared, unless dependent upon a computation

Class: not all variables in the class are initialized where they are declared.

Declarations appear at the beginning of blocks (A block is any code surrounded by curly braces "{" and "}"). The exception is a variable can be declared in a 'for' loop.

F1: Lines 1122,1136,1199,1229,1240,1267 contains variables declarations at the beginning of blocks;
F2: Lines 1334, 1340, 1359, 1372 contains variables declarations at the beginning of blocks;
F3: Line 1533 contains variables declarations at the beginning of blocks;
F4: Line 1597 contains variables declarations at the beginning of blocks.

2.11. Method Calls

Check that parameters are presented in the correct order

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

Check that the correct method is being called, or should it be a different method with a similar name

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

Check that method returned values are used properly

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

2.12. Arrays

Check that there are no off-by-one errors in array indexing (that is, all required array elements are correctly accessed through the index)

F1: No arrays or collections are used;
F2: No arrays or collections are used;
F3: No arrays or collections are used;
F4: No arrays or collections are used.

38. Check that all array (or other collection) indexes have been prevented from going out-of-bounds

F1: No arrays or collections are used;
F2: No arrays or collections are used;
F3: No arrays or collections are used;
F4: No arrays or collections are used.

39. Check that constructors are called when a new array item is desired

F1: No arrays or collections are used;
F2: No arrays or collections are used;
F3: No arrays or collections are used;
F4: No arrays or collections are used.

2.13. Object Comparison

Check that all objects (including Strings) are compared with "equals" and not with "=="

F1: Line 1130,1139,1219,1268 present comparisons with “==” or “!=”;
F2: Line 1336 presents a comparison with “!=”;
F3: Lines 1525, 1536 and1547 present a comparison with “==” or “!=”;
F4: Lines 1592, 1596, 1598, 1603 and 1611 present a comparison with “==” or “!=”.

2.14. Output Format

Check that displayed output is free of spelling and grammatical errors

F1: No errors found;
F2: No errors found;
F3: No errors found;
F4: No errors found.

Check that error messages are comprehensive and provide guidance as to how to correct the problem

F1: All the logger messages explain quite well the errors or how to correct problems;
F2: All the logger messages explain quite well the errors or how to correct problems;
F3: No error messages are present;
F4: All the logger messages explain quite well the errors or how to correct problems.

Check that the output is formatted correctly in terms of line stepping and spacing

F1: No issues;

F2: No issues;
F3: No issues;
F4: No issues.

2.15. Computation, Comparisons and Assignments

Check that the implementation avoids "brutish programming": (see http://users.csc.calpoly.edu/~jdalbey/SWE/CodeSmells/bonehead.html)

F1: Line 1199 it is difficult to read:
`boolean redeliver = (timerState.isExpired())? false : container.callEJBTimeout(timerState, this);`
F2: F2 implementation avoids brutish programming;
F3: F3 implementation avoids brutish programming;
F4: F4 implementation avoids brutish programming.

Check order of computation/evaluation, operator precedence and parenthesizing

F1: It is used only "+" operator and the precedence is respected;
F2: It is used only "+" operator and the precedence is respected;
F3: It is used only "+" operator and the precedence is respected;
F4: It is used only "+" operator and the precedence is respected.

Check the liberal use of parenthesis is used to avoid operator precedence problems.

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

Check that all denominators of a division are prevented from being zero

F1: No division operations are present;
F2: No division operations are present;
F3: No division operations are present;
F4: No division operations are present.

Check that integer arithmetic, especially division, are used appropriately to avoid causing unexpected truncation/rounding

F1: No arithmetic operations present (there is only the sum that does not cause truncations or rounding);
F2: No arithmetic operations present (there is only the sum that does not cause truncations or rounding);
F3: No arithmetic operations present;

F4: No arithmetic operations present.

Check that the comparison and Boolean operators are correct

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

Check throw-catch expressions, and check that the error condition is actually legitimate

F1: No issues;
F2: No issues;
F3: No issues;
F4: No issues.

Check that the code is free of any implicit type conversions

F1: There are no conversions;
F2: There are no conversions;
F3: There are no conversions;
F4: There are no conversions.

2.16. Exceptions

Check that the relevant exceptions are caught

F1: There is only one try-catch and it is relevant;
F2: There is only one try-catch and it is relevant;
F3: There are not try-catch blocks;
F4: In line 1614 an exception is thrown without try-catch block.

Check that the appropriate actions are taken for each catch block

F1: When the try-catch crash the timer is canceled and all the actions are undone;
F2: When the try-catch crash the timer is canceled and all the actions are undone;
F3: No try-catch blocks;
F4: No try-catch blocks.

2.17. Flow of Control

In a switch statement, check that all cases are addressed by break or return

F1: No switch present;
F2: No switch present;

F3: No switch present;
F4: No switch present.

Check that all switch statements have a default branch

F1: No switch present;
F2: No switch present;
F3: No switch present;
F4: No switch present.

Check that all loops are correctly formed, with the appropriate initialization, increment and termination expressions

F1: No loops present;
F2: No loops present;
F3: No loops present;
F4: No loops present.

2.18. Files

Check that all files are properly declared and opened

F1: No file present;
F2: No file present;
F3: No file present;
F4: No file present.

Check that all files are closed properly, even in the case of an error

F1: No file present;
F2: No file present;
F3: No file present;
F4: No file present.

Check that EOF conditions are detected and handled correctly

F1: No file present;
F2: No file present;
F3: No file present;
F4: No file present.

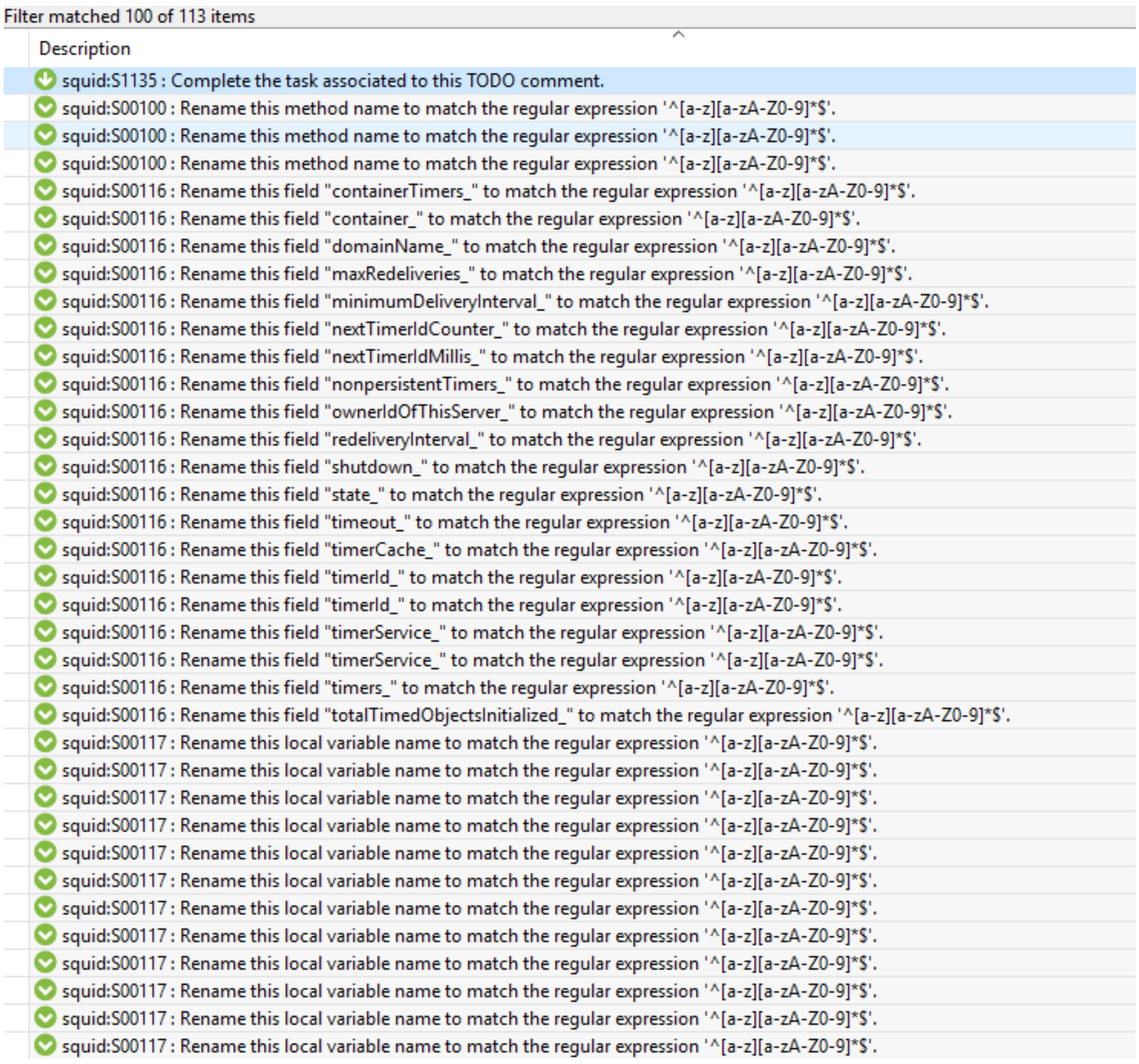
Check that all file exceptions are caught and dealt with accordingly

F1: No file present;
F2: No file present;

F3: No file present;
F4: No file present.

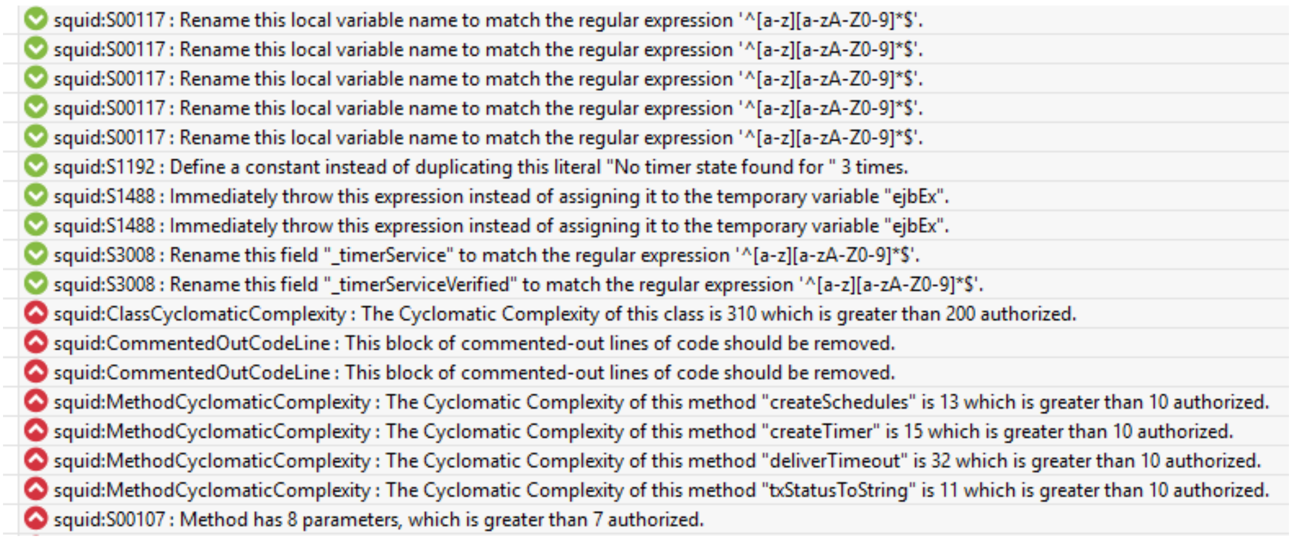
3. Other problems present in the class

We did not found code that might create bugs but there are some parts that may be done better. Here is the result of the analysis done with SonarQube.

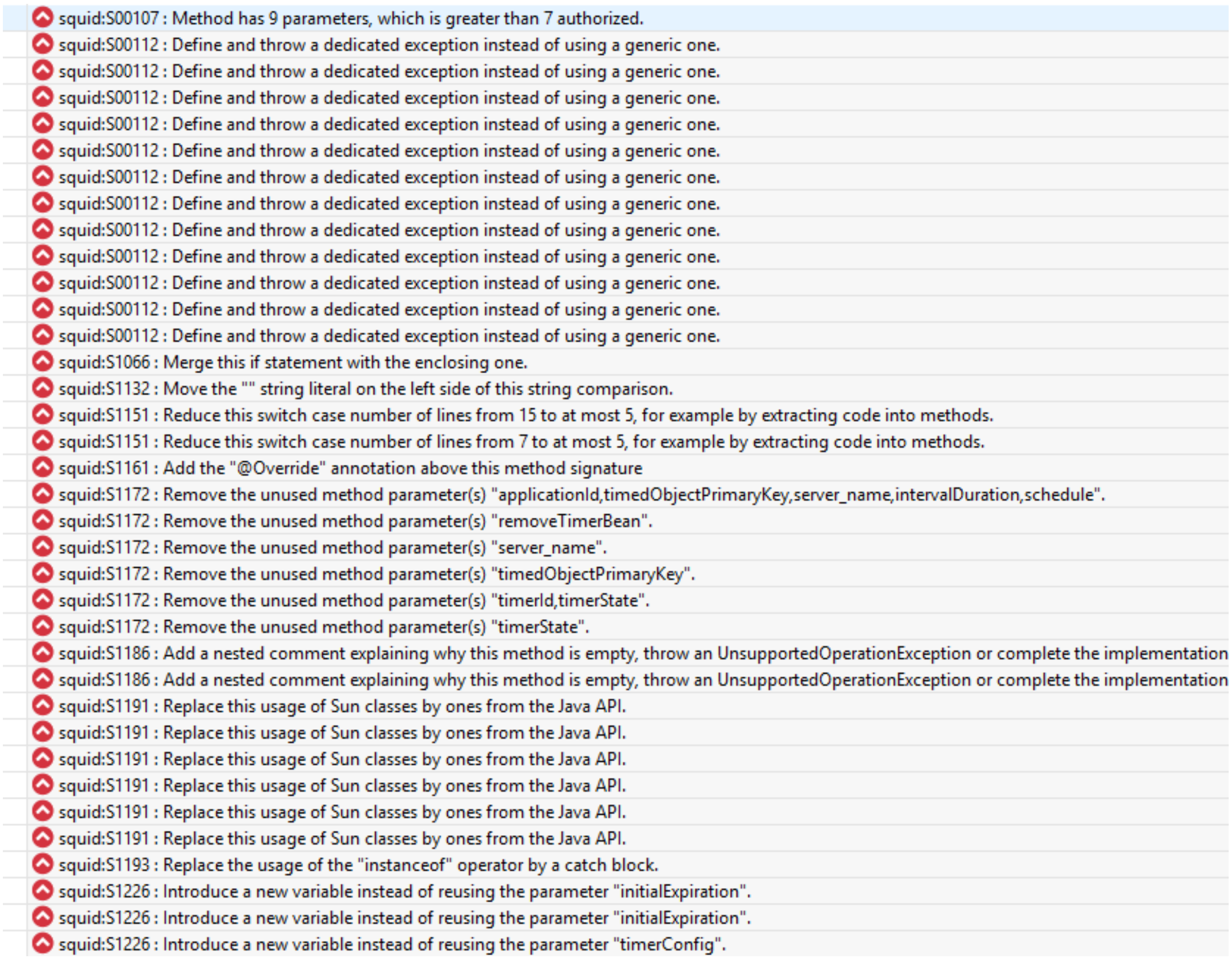


Filter matched 100 of 113 items

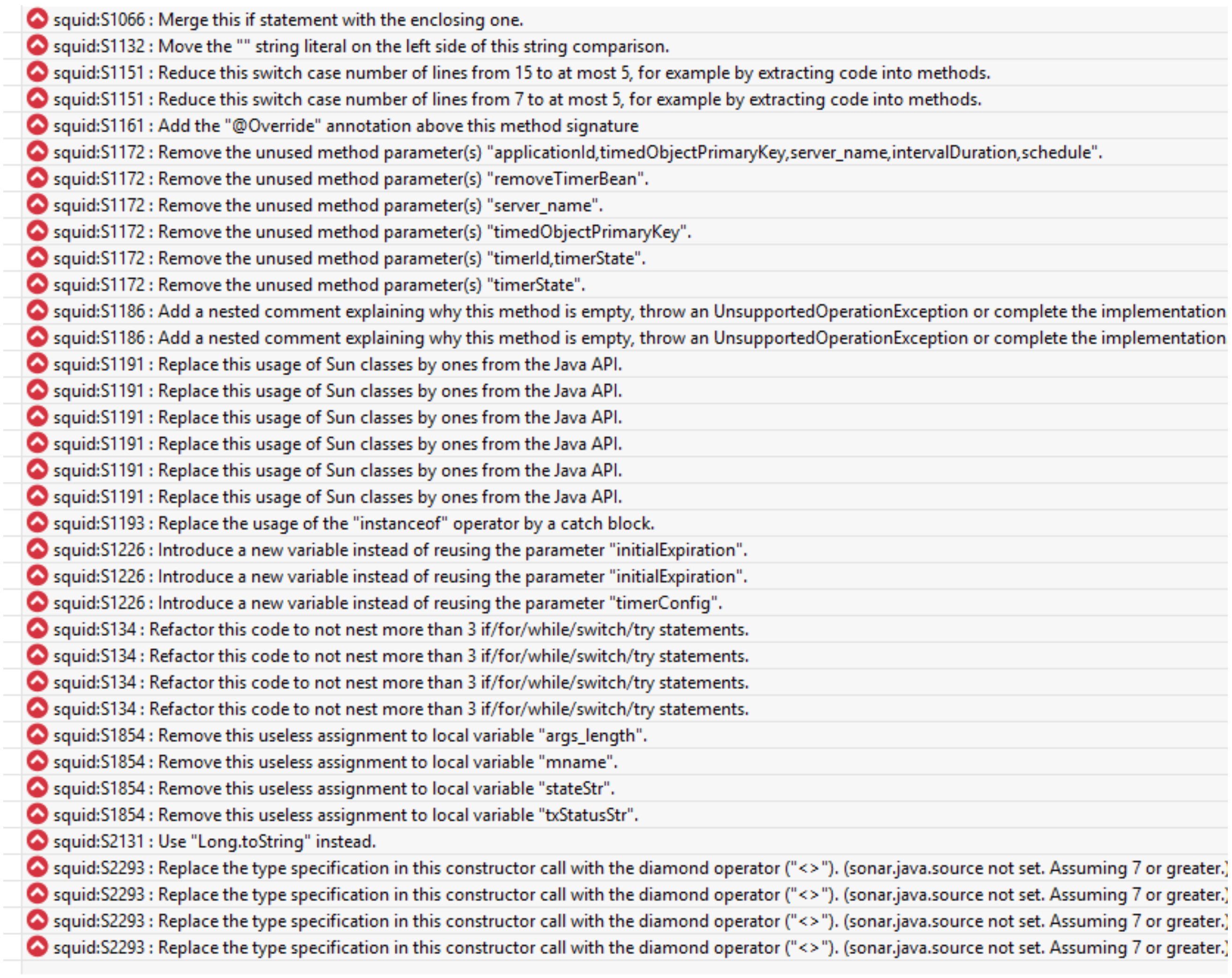
Description
squid:S1135 : Complete the task associated to this TODO comment.
squid:S00100 : Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00100 : Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00100 : Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "containerTimers_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "container_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "domainName_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "maxRedeliveries_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "minimumDeliveryInterval_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "nextTimerIdCounter_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "nextTimerIdMillis_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "nonpersistentTimers_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "ownerIdOfThisServer_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "redeliveryInterval_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "shutdown_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "state_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timeout_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timerCache_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timerId_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timerId_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timerService_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timerService_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "timers_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00116 : Rename this field "totalTimedObjectsInitialized_" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.



```
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S00117 : Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S1192 : Define a constant instead of duplicating this literal "No timer state found for " 3 times.
squid:S1488 : Immediately throw this expression instead of assigning it to the temporary variable "ejbEx".
squid:S1488 : Immediately throw this expression instead of assigning it to the temporary variable "ejbEx".
squid:S3008 : Rename this field "_timerService" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:S3008 : Rename this field "_timerServiceVerified" to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
squid:ClassCyclomaticComplexity : The Cyclomatic Complexity of this class is 310 which is greater than 200 authorized.
squid:CommentedOutCodeLine : This block of commented-out lines of code should be removed.
squid:CommentedOutCodeLine : This block of commented-out lines of code should be removed.
squid:MethodCyclomaticComplexity : The Cyclomatic Complexity of this method "createSchedules" is 13 which is greater than 10 authorized.
squid:MethodCyclomaticComplexity : The Cyclomatic Complexity of this method "createTimer" is 15 which is greater than 10 authorized.
squid:MethodCyclomaticComplexity : The Cyclomatic Complexity of this method "deliverTimeout" is 32 which is greater than 10 authorized.
squid:MethodCyclomaticComplexity : The Cyclomatic Complexity of this method "txStatusToString" is 11 which is greater than 10 authorized.
squid:S00107 : Method has 8 parameters, which is greater than 7 authorized.
```



```
squid:S00107 : Method has 9 parameters, which is greater than 7 authorized.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S00112 : Define and throw a dedicated exception instead of using a generic one.
squid:S1066 : Merge this if statement with the enclosing one.
squid:S1132 : Move the "" string literal on the left side of this string comparison.
squid:S1151 : Reduce this switch case number of lines from 15 to at most 5, for example by extracting code into methods.
squid:S1151 : Reduce this switch case number of lines from 7 to at most 5, for example by extracting code into methods.
squid:S1161 : Add the "@Override" annotation above this method signature
squid:S1172 : Remove the unused method parameter(s) "applicationId,timedObjectPrimaryKey,server_name,intervalDuration,schedule".
squid:S1172 : Remove the unused method parameter(s) "removeTimerBean".
squid:S1172 : Remove the unused method parameter(s) "server_name".
squid:S1172 : Remove the unused method parameter(s) "timedObjectPrimaryKey".
squid:S1172 : Remove the unused method parameter(s) "timerId,timerState".
squid:S1172 : Remove the unused method parameter(s) "timerState".
squid:S1186 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation
squid:S1186 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1193 : Replace the usage of the "instanceof" operator by a catch block.
squid:S1226 : Introduce a new variable instead of reusing the parameter "initialExpiration".
squid:S1226 : Introduce a new variable instead of reusing the parameter "initialExpiration".
squid:S1226 : Introduce a new variable instead of reusing the parameter "timerConfig".
```



```
squid:S1066 : Merge this if statement with the enclosing one.
squid:S1132 : Move the "" string literal on the left side of this string comparison.
squid:S1151 : Reduce this switch case number of lines from 15 to at most 5, for example by extracting code into methods.
squid:S1151 : Reduce this switch case number of lines from 7 to at most 5, for example by extracting code into methods.
squid:S1161 : Add the "@Override" annotation above this method signature
squid:S1172 : Remove the unused method parameter(s) "applicationId,timedObjectPrimaryKey,server_name,intervalDuration,schedule".
squid:S1172 : Remove the unused method parameter(s) "removeTimerBean".
squid:S1172 : Remove the unused method parameter(s) "server_name".
squid:S1172 : Remove the unused method parameter(s) "timedObjectPrimaryKey".
squid:S1172 : Remove the unused method parameter(s) "timerId,timerState".
squid:S1172 : Remove the unused method parameter(s) "timerState".
squid:S1186 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation
squid:S1186 : Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1191 : Replace this usage of Sun classes by ones from the Java API.
squid:S1193 : Replace the usage of the "instanceof" operator by a catch block.
squid:S1226 : Introduce a new variable instead of reusing the parameter "initialExpiration".
squid:S1226 : Introduce a new variable instead of reusing the parameter "initialExpiration".
squid:S1226 : Introduce a new variable instead of reusing the parameter "timerConfig".
squid:S134 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.
squid:S134 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.
squid:S134 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.
squid:S134 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.
squid:S1854 : Remove this useless assignment to local variable "args_length".
squid:S1854 : Remove this useless assignment to local variable "mname".
squid:S1854 : Remove this useless assignment to local variable "stateStr".
squid:S1854 : Remove this useless assignment to local variable "txStatusStr".
squid:S2131 : Use "Long.toString" instead.
squid:S2293 : Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.
squid:S2293 : Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.
squid:S2293 : Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.
squid:S2293 : Replace the type specification in this constructor call with the diamond operator ("<>"). (sonar.java.source not set. Assuming 7 or greater.
```

4. Design Document modifications

Here are explained the modifications that we have added into the DD document:

- We have corrected some little layout mistakes;
- In the Algorithm Design part, we have changed the type object “taxi” from “Object” to “Taxi” in order to be more precise.

5. References

In order to write this document, we have consulted the following bibliography.

- Assignment 3 - Code Inspection.pdf

6. Used Tools

We have used the following programs in order to write this document.

- Netbeans IDE 8.1 for Java EE;
- SonarQube: to analyze the software;
- Microsoft Word 2016 for Mac and Windows: to write and format the document.

7. Hours of work

Each component of the group has worked the following number of hours in order to write this document:

- Elis Bardhi: 6 hours;
- Andrea Cavalli: 5 hours;
- Mario Dolci: 5 hours.