DELHI TECHNOLOGICAL UNIVERSITY

SOFTWARE QUALITY & METRICS (SE - 411)

TEST 02

Submitted To:

Dr. Marouane Kessentini

Submitted By:

Ashish Kumar

2K18/SE/041

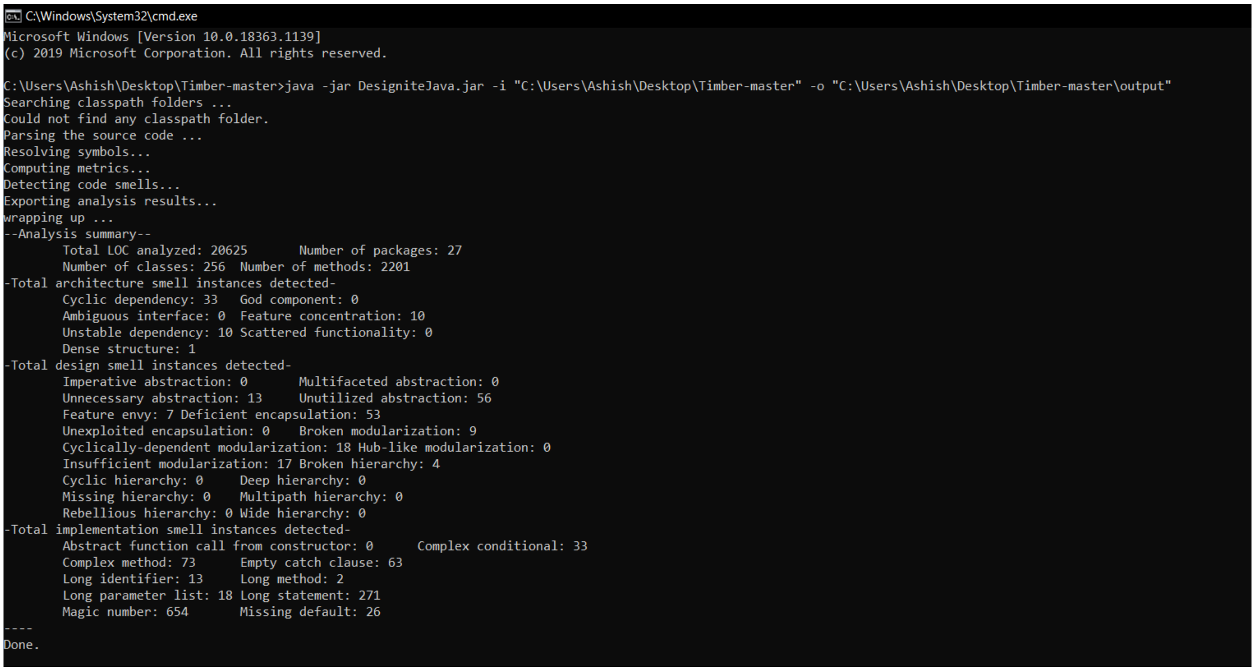
Designite

DesigniteJava is a code quality assessment tool for code written in Java. It detects numerous architecture, design, and implementation smells that show maintainability issues present in the analyzed code. It also computes many commonly used object-oriented metrics. It helps us **reduce technical debt and improve maintainability** of our software.

Steps to execute the chosen project on DesigniteJava

- I have chosen a **timber named project** from github and it contains more than 5K LOC (Line of Code).
- Then I download the DesigniteJava.jar file from the Designite website (https://www.designite-tools.com/designitejava/).
- After that I download that project on my computer and paste the jar file into that folder where I have download the project and then open the terminal and executes following command:

  java -jar DesigniteJava.jar -i "C:\Users\Ashish\Desktop\Timber-master" -o "C:\Users\Ashish\Desktop\Timber-master\output"
- After executing the command successfully, results are created in output folder and following output is displayed on the console:



```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Ashish\Desktop\Timber-master>java -jar DesigniteJava.jar -i "C:\Users\Ashish\Desktop\Timber-master" -o "C:\Users\Ashish\Desktop\Timber-master\output"
Searching classpath folders ...
Could not find any classpath folder.
Parsing the source code ...
Resolving symbols...
Computing metrics...
Detecting code smells...
Exporting analysis results...
wrapping up ...
--Analysis summary--
        Total LOC analyzed: 20625       Number of packages: 27
        Number of classes: 256  Number of methods: 2201
-Total architecture smell instances detected-
        Cyclic dependency: 33   God component: 0
        Ambiguous interface: 0  Feature concentration: 10
        Unstable dependency: 10 Scattered functionality: 0
        Dense structure: 1
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 0
        Unnecessary abstraction: 13     Unutilized abstraction: 56
        Feature envy: 7 Deficient encapsulation: 53
        Unexploited encapsulation: 0    Broken modularization: 9
        Cyclically-dependent modularization: 18 Hub-like modularization: 0
        Insufficient modularization: 17 Broken hierarchy: 4
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 33
        Complex method: 73      Empty catch clause: 63
        Long identifier: 13     Long method: 2
        Long parameter list: 18 Long statement: 271
        Magic number: 654       Missing default: 26
----
Done.
```

- Google drive Link of the results (in csv format) that are generated after executing the above command:
  https://drive.google.com/drive/folders/1Trj3GphTJ_Wp1QD1BQGlKgX7k4gtjGf5?usp=sharing

RESULTS

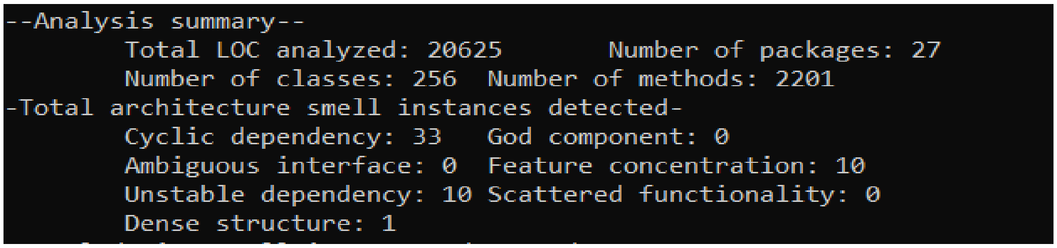
Types of Code Smells

1. Architecture smells

It is defined as an initial set of architecture smells including ambiguous interface and scattered parasitic functionality. Garcia provides mathematically formal definitions of smells that help implement the smell detection tools for architecture smells. It presents an extended catalog of architectural smells along with the impacted quality attributes.

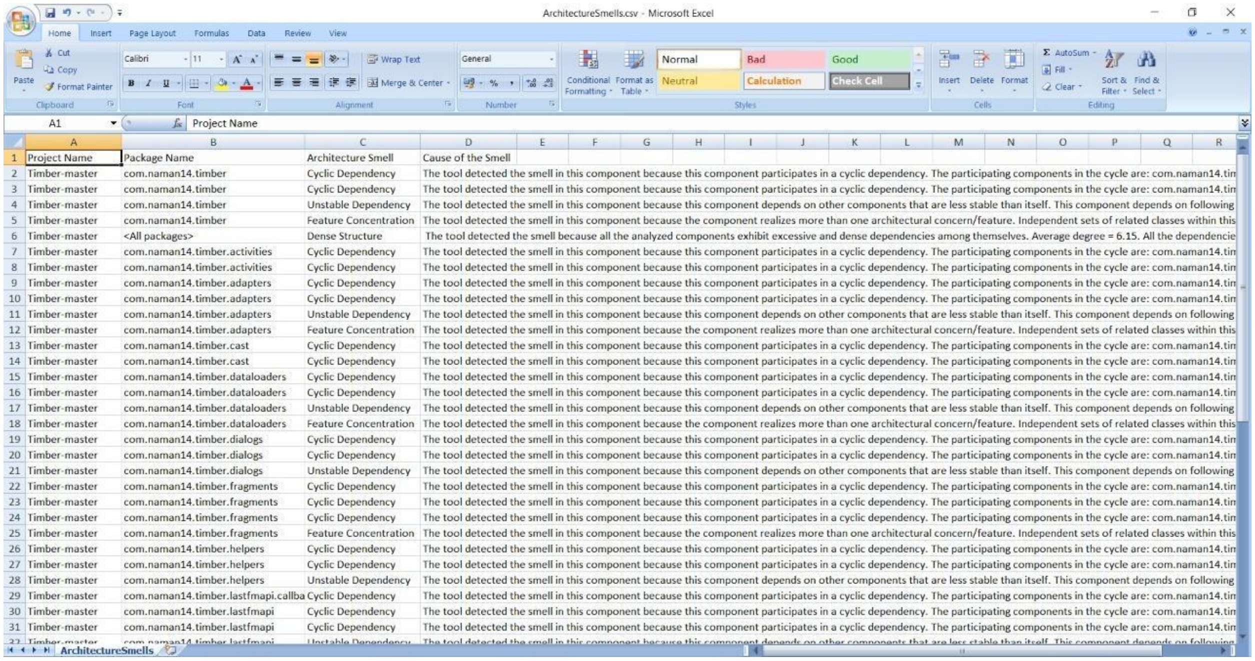
I have detected following instances of architecture smells:

There are 33 cyclic dependency, 10 unstable dependency, 10 feature concentration and 1 dense structure.



```
--Analysis summary--
        Total LOC analyzed: 20625       Number of packages: 27
        Number of classes: 256  Number of methods: 2201
-Total architecture smell instances detected-
        Cyclic dependency: 33   God component: 0
        Ambiguous interface: 0  Feature concentration: 10
        Unstable dependency: 10 Scattered functionality: 0
        Dense structure: 1
```

And following screenshot is of the results of architecture smells in csv format:



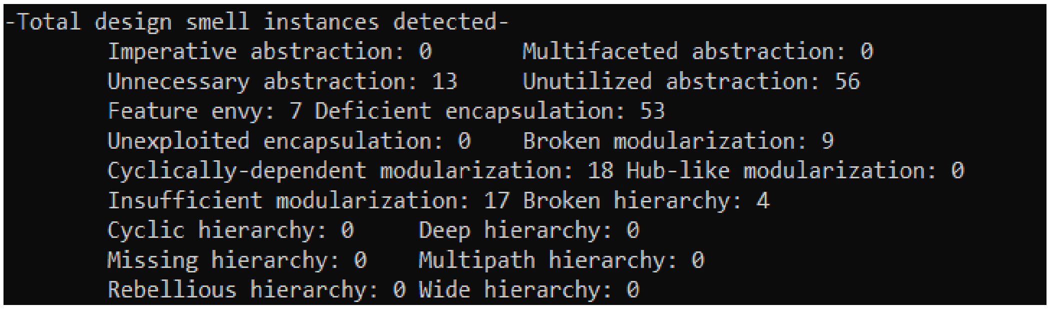
Project Name	Package Name	Architecture Smell	Cause of the Smell
Timber-master	com.naman14.timber	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following
Timber-master	com.naman14.timber	Feature Concentration	The tool detected the smell in this component because the component realizes more than one architectural concern/feature. Independent sets of related classes within this
Timber-master	<All packages>	Dense Structure	The tool detected the smell because all the analyzed components exhibit excessive and dense dependencies among themselves. Average degree = 6.15. All the dependencie
Timber-master	com.naman14.timber.activities	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.activities	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.adapters	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.adapters	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.adapters	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following
Timber-master	com.naman14.timber.adapters	Feature Concentration	The tool detected the smell in this component because the component realizes more than one architectural concern/feature. Independent sets of related classes within this
Timber-master	com.naman14.timber.cast	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.cast	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.dataloaders	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.dataloaders	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.dataloaders	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following
Timber-master	com.naman14.timber.dataloaders	Feature Concentration	The tool detected the smell in this component because the component realizes more than one architectural concern/feature. Independent sets of related classes within this
Timber-master	com.naman14.timber.dialogs	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.dialogs	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.dialogs	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following
Timber-master	com.naman14.timber.fragments	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.fragments	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.fragments	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.fragments	Feature Concentration	The tool detected the smell in this component because the component realizes more than one architectural concern/feature. Independent sets of related classes within this
Timber-master	com.naman14.timber.helpers	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.helpers	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.helpers	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following
Timber-master	com.naman14.timber.lastfmapi.callba	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.lastfmapi	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.lastfmapi	Cyclic Dependency	The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.naman14.tim
Timber-master	com.naman14.timber.lastfmapi	Unstable Dependency	The tool detected the smell in this component because this component depends on other components that are less stable than itself. This component depends on following

2. Design smells

This is a prominent technique used to detect smells are metrics-based, rules-based (or heuristics), machine learning base, and history-based. However, most of the present tools support the detection of a very scanty kind of smells and target mainly Java programming language. In computer programming, **design smells** are "structures in the design that indicate violation of fundamental design principles and negatively impact design quality”.

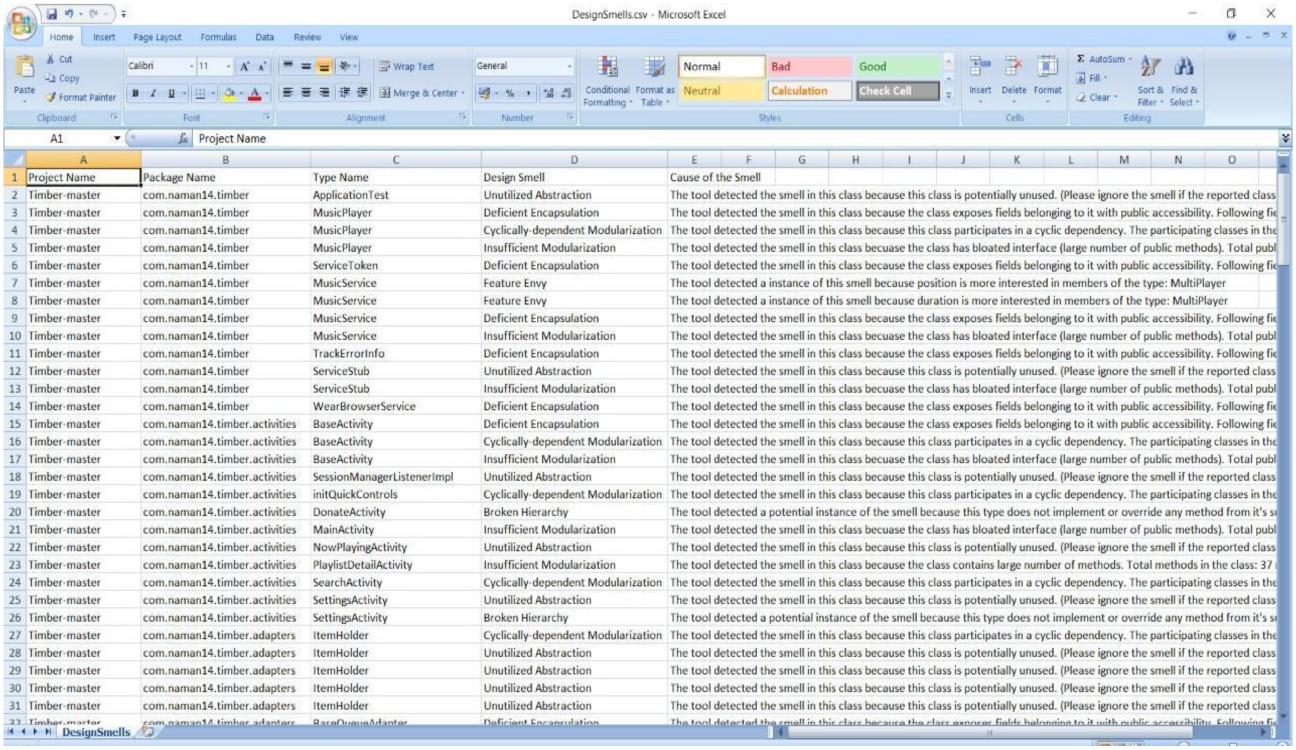
I have detected following instances of design smells:

There are 13 Unnecessary abstraction, 56 Unutilized abstraction, 7 feature envy, 53 deficient encapsulation, 9 broken modularization, 18 cyclically-dependent modularization and many more which are listed below:



```
-Total design smell instances detected-
        Imperative abstraction: 0       Multifaceted abstraction: 0
        Unnecessary abstraction: 13     Unutilized abstraction: 56
        Feature envy: 7 Deficient encapsulation: 53
        Unexploited encapsulation: 0    Broken modularization: 9
        Cyclically-dependent modularization: 18 Hub-like modularization: 0
        Insufficient modularization: 17 Broken hierarchy: 4
        Cyclic hierarchy: 0     Deep hierarchy: 0
        Missing hierarchy: 0    Multipath hierarchy: 0
        Rebellious hierarchy: 0 Wide hierarchy: 0
```

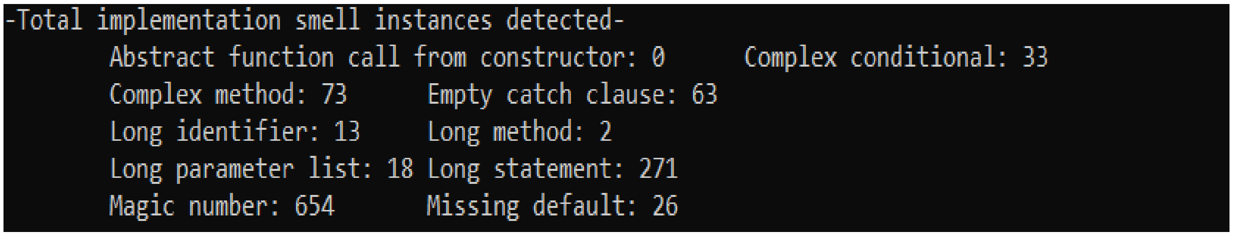
Following screenshot is of the results of design smells in csv format:



	Project Name	Package Name	Type Name	Design Smell	Cause of the Smell
2	Timber-master	com.naman14.timber	ApplicationTest	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
3	Timber-master	com.naman14.timber	MusicPlayer	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
4	Timber-master	com.naman14.timber	MusicPlayer	Cyclically-dependent Modularization	The tool detected the smell in this class because this class participates in a cyclic dependency. The participating classes in the
5	Timber-master	com.naman14.timber	MusicPlayer	Insufficient Modularization	The tool detected the smell in this class becuase the class has bloated interface (large number of public methods). Total publ
6	Timber-master	com.naman14.timber	ServiceToken	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
7	Timber-master	com.naman14.timber	MusicService	Feature Envy	The tool detected a instance of this smell because position is more interested in members of the type: MultiPlayer
8	Timber-master	com.naman14.timber	MusicService	Feature Envy	The tool detected a instance of this smell because duration is more interested in members of the type: MultiPlayer
9	Timber-master	com.naman14.timber	MusicService	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
10	Timber-master	com.naman14.timber	MusicService	Insufficient Modularization	The tool detected the smell in this class becuase the class has bloated interface (large number of public methods). Total publ
11	Timber-master	com.naman14.timber	TrackErrorInfo	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
12	Timber-master	com.naman14.timber	ServiceStub	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
13	Timber-master	com.naman14.timber	ServiceStub	Insufficient Modularization	The tool detected the smell in this class becuase the class has bloated interface (large number of public methods). Total publ
14	Timber-master	com.naman14.timber	WearBrowserService	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
15	Timber-master	com.naman14.timber.activities	BaseActivity	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fie
16	Timber-master	com.naman14.timber.activities	BaseActivity	Cyclically-dependent Modularization	The tool detected the smell in this class because this class participates in a cyclic dependency. The participating classes in the
17	Timber-master	com.naman14.timber.activities	BaseActivity	Insufficient Modularization	The tool detected the smell in this class becuase the class has bloated interface (large number of public methods). Total publ
18	Timber-master	com.naman14.timber.activities	SessionManagerListenerImpl	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
19	Timber-master	com.naman14.timber.activities	initQuickControls	Cyclically-dependent Modularization	The tool detected the smell in this class because this class participates in a cyclic dependency. The participating classes in the
20	Timber-master	com.naman14.timber.activities	DonateActivity	Broken Hierarchy	The tool detected a potential instance of the smell because this type does not implement or override any method from it's s
21	Timber-master	com.naman14.timber.activities	MainActivity	Insufficient Modularization	The tool detected the smell in this class becuase the class has bloated interface (large number of public methods). Total publ
22	Timber-master	com.naman14.timber.activities	NowPlayingActivity	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
23	Timber-master	com.naman14.timber.activities	PlaylistDetailActivity	Insufficient Modularization	The tool detected the smell in this class because the class contains large number of methods. Total methods in the class: 37
24	Timber-master	com.naman14.timber.activities	SearchActivity	Cyclically-dependent Modularization	The tool detected the smell in this class because this class participates in a cyclic dependency. The participating classes in the
25	Timber-master	com.naman14.timber.activities	SettingsActivity	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
26	Timber-master	com.naman14.timber.activities	SettingsActivity	Broken Hierarchy	The tool detected a potential instance of the smell because this type does not implement or override any method from it's s
27	Timber-master	com.naman14.timber.adapters	ItemHolder	Cyclically-dependent Modularization	The tool detected the smell in this class because this class participates in a cyclic dependency. The participating classes in the
28	Timber-master	com.naman14.timber.adapters	ItemHolder	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
29	Timber-master	com.naman14.timber.adapters	ItemHolder	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
30	Timber-master	com.naman14.timber.adapters	ItemHolder	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
31	Timber-master	com.naman14.timber.adapters	ItemHolder	Unutilized Abstraction	The tool detected the smell in this class because this class is potentially unused. (Please ignore the smell if the reported class
32	Timber-master	com.naman14.timber.adapters	BaseQueueAdapter	Deficient Encapsulation	The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fi

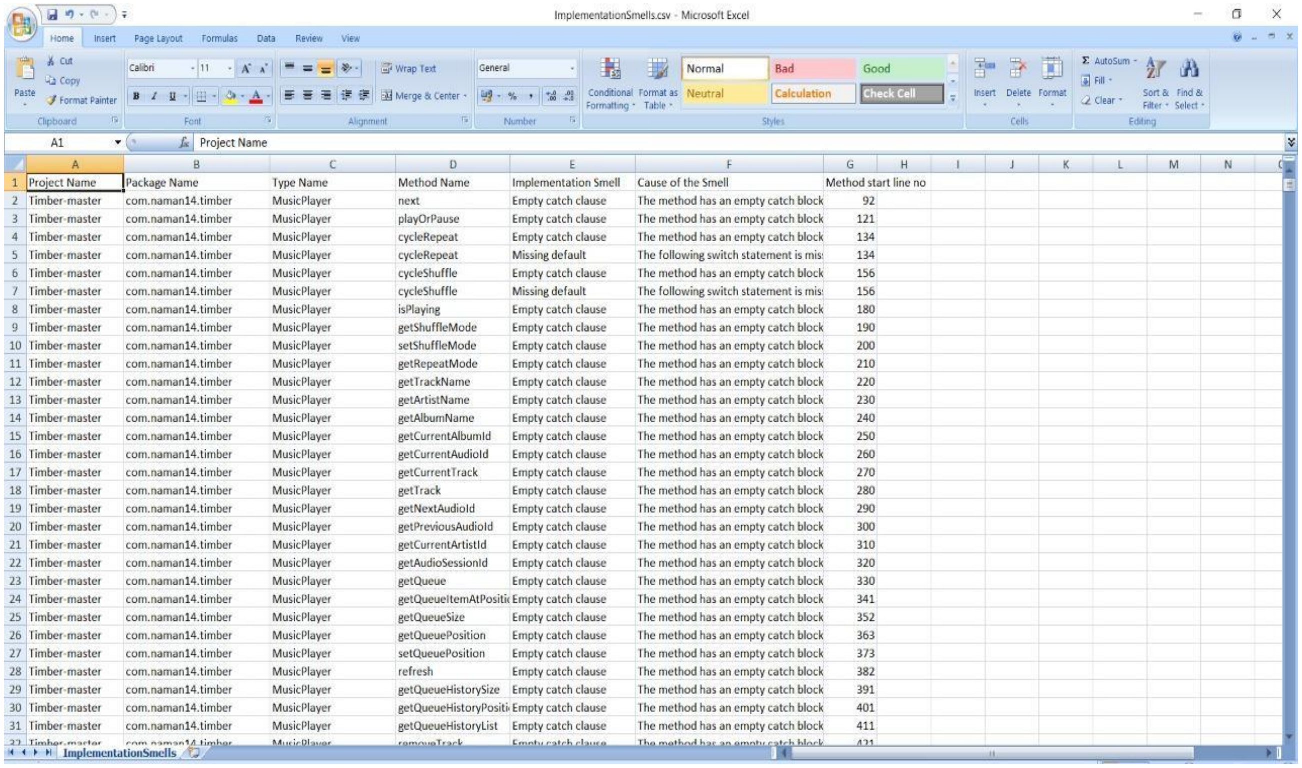
3. Implementation smells

I have detected following instances of implementation smells:
There are 33 complex conditional, 73 complex methods, 63 empty clause, 13 long identifier, 2 long method, 654 magic number and 26 missing default.



```
-Total implementation smell instances detected-
        Abstract function call from constructor: 0      Complex conditional: 33
        Complex method: 73      Empty catch clause: 63
        Long identifier: 13     Long method: 2
        Long parameter list: 18 Long statement: 271
        Magic number: 654       Missing default: 26
```

Following screenshot is of the results of implementation smells in csv format:



A1 | Project Name

	Project Name	Package Name	Type Name	Method Name	Implementation Smell	Cause of the Smell	Method start line no
2	Timber-master	com.naman14.timber	MusicPlayer	next	Empty catch clause	The method has an empty catch block	92
3	Timber-master	com.naman14.timber	MusicPlayer	playOrPause	Empty catch clause	The method has an empty catch block	121
4	Timber-master	com.naman14.timber	MusicPlayer	cycleRepeat	Empty catch clause	The method has an empty catch block	134
5	Timber-master	com.naman14.timber	MusicPlayer	cycleRepeat	Missing default	The following switch statement is mis:	134
6	Timber-master	com.naman14.timber	MusicPlayer	cycleShuffle	Empty catch clause	The method has an empty catch block	156
7	Timber-master	com.naman14.timber	MusicPlayer	cycleShuffle	Missing default	The following switch statement is mis:	156
8	Timber-master	com.naman14.timber	MusicPlayer	isPlaying	Empty catch clause	The method has an empty catch block	180
9	Timber-master	com.naman14.timber	MusicPlayer	getShuffleMode	Empty catch clause	The method has an empty catch block	190
10	Timber-master	com.naman14.timber	MusicPlayer	setShuffleMode	Empty catch clause	The method has an empty catch block	200
11	Timber-master	com.naman14.timber	MusicPlayer	getRepeatMode	Empty catch clause	The method has an empty catch block	210
12	Timber-master	com.naman14.timber	MusicPlayer	getTrackName	Empty catch clause	The method has an empty catch block	220
13	Timber-master	com.naman14.timber	MusicPlayer	getArtistName	Empty catch clause	The method has an empty catch block	230
14	Timber-master	com.naman14.timber	MusicPlayer	getAlbumName	Empty catch clause	The method has an empty catch block	240
15	Timber-master	com.naman14.timber	MusicPlayer	getCurrentAlbumId	Empty catch clause	The method has an empty catch block	250
16	Timber-master	com.naman14.timber	MusicPlayer	getCurrentAudioId	Empty catch clause	The method has an empty catch block	260
17	Timber-master	com.naman14.timber	MusicPlayer	getCurrentTrack	Empty catch clause	The method has an empty catch block	270
18	Timber-master	com.naman14.timber	MusicPlayer	getTrack	Empty catch clause	The method has an empty catch block	280
19	Timber-master	com.naman14.timber	MusicPlayer	getNextAudioId	Empty catch clause	The method has an empty catch block	290
20	Timber-master	com.naman14.timber	MusicPlayer	getPreviousAudioId	Empty catch clause	The method has an empty catch block	300
21	Timber-master	com.naman14.timber	MusicPlayer	getCurrentArtistId	Empty catch clause	The method has an empty catch block	310
22	Timber-master	com.naman14.timber	MusicPlayer	getAudioSessionId	Empty catch clause	The method has an empty catch block	320
23	Timber-master	com.naman14.timber	MusicPlayer	getQueue	Empty catch clause	The method has an empty catch block	330
24	Timber-master	com.naman14.timber	MusicPlayer	getQueueItemAtPositic	Empty catch clause	The method has an empty catch block	341
25	Timber-master	com.naman14.timber	MusicPlayer	getQueueSize	Empty catch clause	The method has an empty catch block	352
26	Timber-master	com.naman14.timber	MusicPlayer	getQueuePosition	Empty catch clause	The method has an empty catch block	363
27	Timber-master	com.naman14.timber	MusicPlayer	setQueuePosition	Empty catch clause	The method has an empty catch block	373
28	Timber-master	com.naman14.timber	MusicPlayer	refresh	Empty catch clause	The method has an empty catch block	382
29	Timber-master	com.naman14.timber	MusicPlayer	getQueueHistorySize	Empty catch clause	The method has an empty catch block	391
30	Timber-master	com.naman14.timber	MusicPlayer	getQueueHistoryPositi	Empty catch clause	The method has an empty catch block	401
31	Timber-master	com.naman14.timber	MusicPlayer	getQueueHistoryList	Empty catch clause	The method has an empty catch block	411
32	Timber-master	com.naman14.timber	MusicPlayer	removeTrack	Empty catch clause	The method has an empty catch block	421

Some MethodMetrics and TypeMetrics has also been generated and that has been provided in Google drive link