



Team Name

Code Breakers

Problem statement title


Method - Trace Analyser

Role of each Team Member

1. Anupriya Singh:

- Primarily involved on working on the, goal, “Compare two trace files: one from failing and passing case each and find out the anomaly”.
- Used Xtrace commands to generate trace files (.trc) and convert it into log files with the help of Command Prompt.
- Developed a java code to read the log files line by line and determined whether or not the exception is present or not.
- Displayed anomalies in the stack trace files
- Integration of all the modules into single code
- Managed and coordinated the working of other members

2. Jyoti Pandey


- Designing a Graphical User Interface(GUI) for Method Tracer using WindowBuilder
 - The GUI consists of text fields etc and is the part where user can interact.
 - Integration of all the modules into single code.
 - Worked on writing code to print commands directly in the command prompt.
 - Worked on running the appropriate java file from the GUI
- 

Role of each team member cont

3. Nidhi Wadhwa

- Primarily worked towards the goal, “Parse one or more trace files and create a tabular view for the number of times each method is invoked”. Calculated the count of number of times a method is invoked in class and displayed the same in a tabular view
- Helped in calculating and displaying entry time , exit time and execution time of each method in the Java file
- Further, displayed which of passing or failing case takes longer to execute and by how much

4. Ronika Das

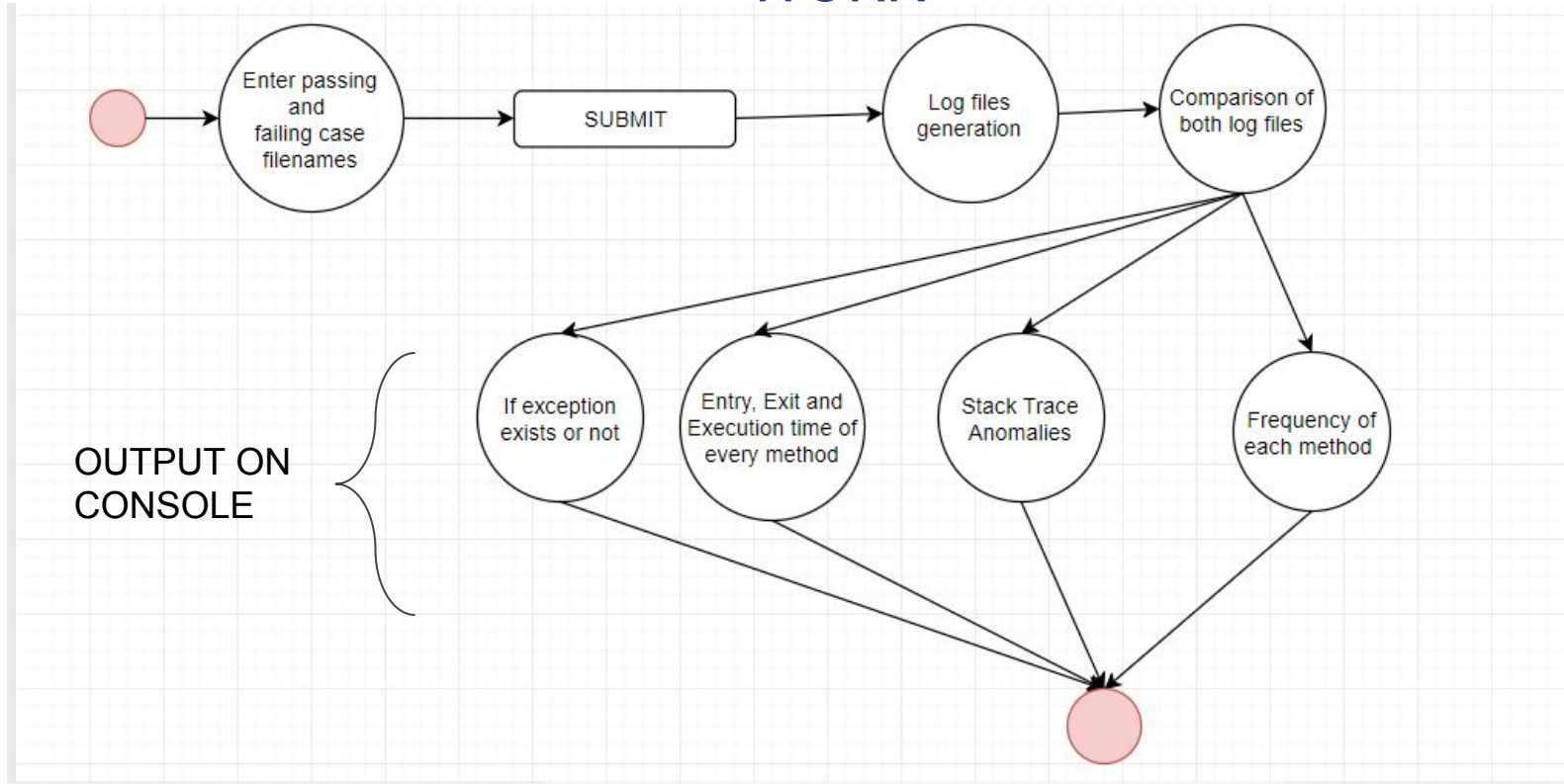
- Involved in generating stack trace of Java files and suggesting ideas to implement the same
 - Suggested ideas and helped in implementing the module to detect exception /anomalies in the generated log files
 - Integration of all modules into a single code
- 

Tools used for Method-Trace Analyser

- Eclipse Photon
- WindowBuilder
- Open JDK with OpenJ9

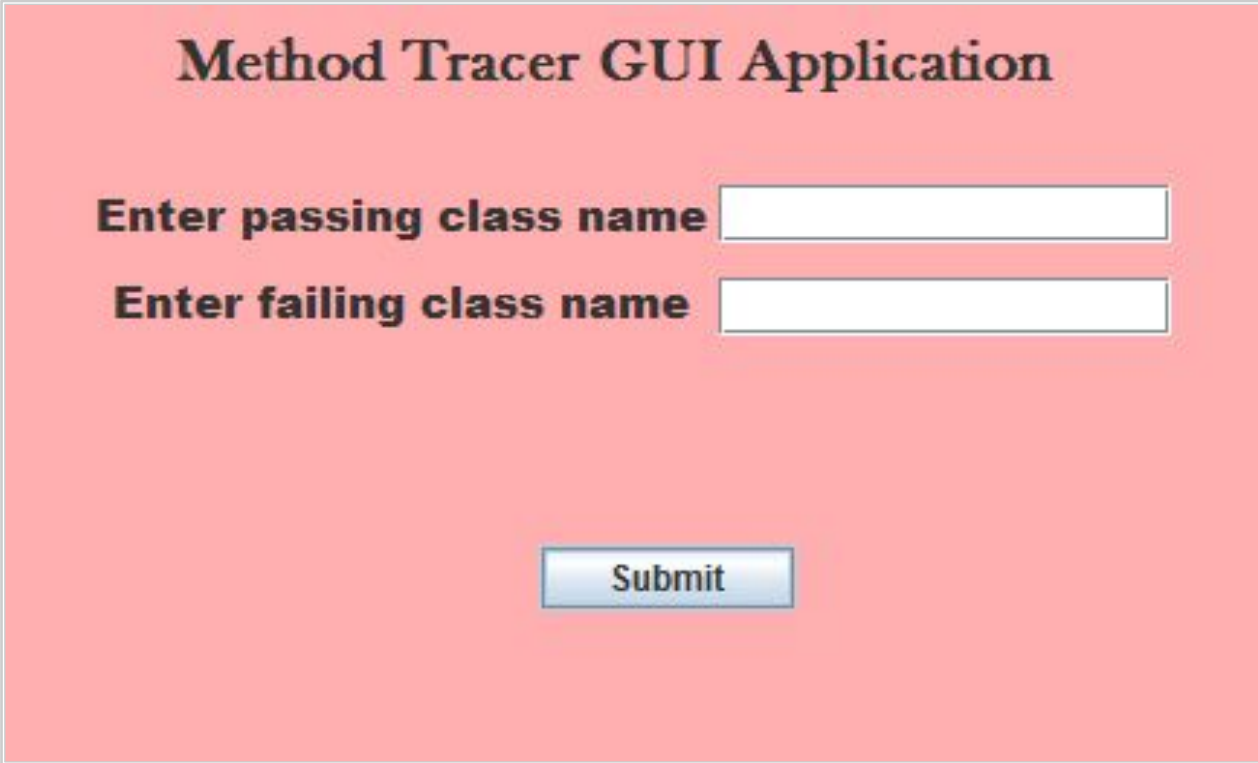


IMPLEMENTED COMPONENTS DEFINED FROM SCOPE OF WORK



FLOW CHART OF THE PROCESS

SCREENSHOT OF GUI



Method Tracer GUI Application

Enter passing class name

Enter failing class name

The screenshot shows a GUI application titled "Method Tracer GUI Application". It has a light red background. There are two text input fields: the first is labeled "Enter passing class name" and the second is labeled "Enter failing class name". Below these fields is a blue "Submit" button.

SCREENSHOTS OF OUTPUT OF METHOD TRACE ANALYSIS

```
LogAnalysis.java Problems Javadoc Declaration Console
<terminated> LogAnalysis [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (02-Oct-2018, 12:2)
Exception exists
=====
Anomalies in jstacktrace for files
  World.java and WorldEx.java :

message() (World.java:15)
=====
Passing case

message
Entry Time = 18:56:41.043980339
Exit Time = 18:56:41.044022659
Execution Time = 4.2319996282458305E-5 seconds

display
Entry Time = 18:56:41.043980339
Exit Time = 18:56:41.044023325
Execution Time = 4.298599378671497E-5 seconds

main
Entry Time = 18:56:41.043980339
Exit Time = 18:56:41.044023658
Execution Time = 4.331899981480092E-5 seconds
```

Failing case

```
display
Entry Time = 18:56:41.585294590
Exit Time = 18:56:41.585951388
Execution Time = 6.567980017280206E-4 seconds
```

```
main
Entry Time = 18:56:41.585294590
Exit Time = 18:56:41.585952850
Execution Time = 6.582600035471842E-4 seconds
```

Passing case takes 6.149410037323833E-4 seconds more to execute.

TraceEntry	Count

display()	1
main()	1
message()	1

GUI

- The GUI component is build using WindowBuilder.
- It consists of two text fields for the class names to be compared and analysed.
- It has a submit button which when clicked will call java method to perform the following tasks:
 - Comparing the trace files, stack trace of passing and failing case for finding out the anomalies.
 - Finding out the time taken by each method for execution.
 - Showing the number of times each method is invoked in the program, in tabular form.
- The output of the tasks is shown on the console.

WORKING

- Firstly, the user needs to enter two class files' names in the given text fields. These should be the ones specifying the passing and failing case of each program.
- Press the submit button.
- This will lead to execution of the program in the background.
- The file names entered will be captured as strings in the code.
- Then, the Xtrace commands will generate trace files(.trc) for the entered files.
- After that, the command to generate the log files from the trace files will be implemented.

WORKING CONT...

- Log files of the passing and failing case are compared to determine what anomalies(exceptions) led to the failing of the program.
- Stack trace is compared to determine when and which line from the code leads to the exception.
- Time taken by each method in the class file to execute is shown, thus helping to know which method took more time.
- And lastly it will also give the count of number of times each method of the program is invoked.
- Output is then displayed on the console.

THE END