

Table 1.1: Actual Input, Output and Variability Information of PLeTs

Functionalities of Functional Testing			
ID	Input	ID	Output
e	Choose a functionality on the menu	5	Functionality chosen
f	Click on File button	6	File options on the screen
g	Click on Configuration button	7	Configuration options on the screen
h	Click on Log button	8	Will be open two options
i	Click on Close button	9	Program closed
j	Click on Edit Configuration File button	10	File configured
k	Log on the screen cleared	11	Click on Clear button
l	Click on Save log file button	12	Log on the screen saved
m	Application will close	13	Application ended
Functionalities of Performance Testing			
ID	Input	ID	Output
n	Open the tool	14	Functionality chosen
o	Click on import XMI/XML file button	15	File exported
p	Click on Help button	16	Help screen opened
q	Click on Environment button	17	Load Runner Path screen opened
r	Click on file button	18	Will be open two options on the screen
s	Click on Parsed Load Runner Script to XMI button	19	Directory screen opened
t	Click on Generate ATC button	20	Convert test data into generic test scenarios
u	Click on import XMI/XML file button	21	File exported
v	Click on Environment button	22	Environment screen opened
x	Choose a Load Runner path   Click on OK button	23	Path chosen
y	Click on Parameterization button	24	Configure data parameterization
z	Click on Generate Scripts button	25	Convert abstract test cases
w	Click on Execute Test button   Click on OK button	26	Invoke the executable program to run the scripts generated
aa	Click on Exit button	27	Program closed
Functional Testing			
ID	Input	ID	Output
ac	(Req->Functionalities Functional.)Click on Load from XMI File button, select file and click on open	29	File XML loaded
ad	Select the method of test sequences generation, HSI	30	File XML loaded
ae	Select the method of test sequences generation, W	31	Method of test sequence generation W is selected
af	Select the method of test sequences generation, Wp	32	Method of test sequence generation Wp is selected
ag	Select the method of test sequences generation, DFS	33	Method of test sequence generation DFS is selected
ah	Click on generate test case from load test data button	34	The abstract test case are generated using HSI method
ai	Click on generate test case from load test data button	35	The abstract test case are generated using W method
aj	Click on generate test case from load test data button	36	The abstract test case are generated using Wp method
ak	Click on generate test case from load test data button	37	The abstract test case are generated using DFS method
al	Click on export file to Visual Studio	38	File exported to Visual Studio
am	File exported to OATS	39	Click on export file to OATS
an	File exported to MTM	40	Click on export file to MTM
ao	File exported to JMeter	41	Export file to Jmeter
ap	Select directory to save   Click on OK button	42	Script on VS saved
aq	Select directory to save   Click on OK button	43	Script on OATS saved
ar	Select directory to save   Click on OK button	44	Script on MTM saved
as	Select directory to save   Click on OK button	45	Script on JMeter saved
at	Click on Load File to be parsed   Select file  Click on Open	46	File XML loaded
au	Choose between close the application or run the application again	47	Command chosen
Choose the Testing Type			
ID	Input	ID	Output
a	Select the type of the test	1	Type of the test is selected
b	Start Functional Testing	2	Functional Testing is started
c	Start Performance Testing	3	Performance Testing is started

d	Start Structural Testing	4	Structural Testing is started
<b>Parameterization</b>			
<b>ID</b>	<b>Input</b>	<b>ID</b>	<b>Output</b>
av	Click on parameterization button	48	The screen will open
ax	Choose the scenario on the left	49	The scenario will be marked on the screen
az	Choose the file on the right	50	The file preview will open on the right
aw	Click on Export scripts file	51	Will be open a screen to choose a directory
ba	Click on OK button to save the script	52	Script is saved
<b>Performance Testing</b>			
<b>ID</b>	<b>Input</b>	<b>ID</b>	<b>Output</b>
bb	(Req->Functionalities Performance)Click on import XMI/XML file button	53	File exported
bc	Click on Generate ATC for HSI button	54	Convert test data into generic test scenarios
bd	Click on Generate ATC for DFS button	55	Convert test data into generic test scenarios
be	Click on Generate ATC for W button	56	Convert test data into generic test scenarios
bf	Click on Generate ATC for WP button	57	Convert test data into generic test scenarios
bg	Click on Parameterization button	58	Configure the data of parameterization
bh	Click on Parameterization button	59	Configure the data of parameterization
bi	Click on Parameterization button	60	Configure the data of parameterization
bj	Click on Parameterization button	61	Configure the data of parameterization
bk	Click on Generate Scripts for Load Runner button	62	Script are generated
bl	Click on Generate Scripts for Visual Studio button	63	Script are generated
bm	Click on Generate Scripts for JMeter button	64	Script are generated
bn	Click on Execute Test button	65	Open Load Runner application to run the scripts generated
bn	Click on Execute Test button	66	Open Visual Studio application to run the scripts generated
bn	Click on Execute Test button	67	Open JMeter application to run the scripts generated
bo	Click on Close button	68	Application closed
<b>Structural Testing</b>			
<b>ID</b>	<b>Input</b>	<b>ID</b>	<b>Output</b>
bs	Type the path of the XMI file on console	72	File XMI loaded
bt	Press Enter	73	Information necessary extracted for generating a data structure in memory
bu	Specify the directory to save the Abstract Structure	74	Directory where is saved the abstract data structure is displayed on the console
bt	Press Enter	86	Data File and Abstract Structure saved
bv	Inform the launcher path of Jabuti, EMMA or Poketool	88	Path informed
bq	Click on Jabuti application located on "c:/PletsCoverageJabutti.exe" (Req->Specify the directory where the Jabuti.jbt file will be stored)	70	Jabuti application opened
br	Click on Emma application located on "c:/PletsCoverageEmma.exe" (Ex->Specify the directory where the Jabuti.jbt file will be stored)	87	PokeTool application opened
ci	Click on Poke-Tool application located on "c:/Poketool.exe" (Ex->Specify the directory where the Jabuti.jbt file will be stored)	71	Emma application opened
cj	Press Enter to export file to JaBUTi	87	Java class for Jabuti is saved
ck	Java class for Poketool is saved	89	Java class for Poketool is saved
cl	Java class for Emma is saved	90	Java class for Emma is saved
ca	Specify the directory where the Jabuti.jbt file will be stored	79	JaBUTi's GUI is launched
cb	Application will open on screen	80	Tests results on screen
cd	Press on Close	81	Application closed
ch	Press Y in order to run the tests again	85	Console ready to export XML file
cj	Select directory to save java class for JaBUTi	87	Java class is saved
ck	Select directory to save java class for poketool	87	Java class is saved
cl	Select directory to save java class for Emma	87	Java class is saved