Assignment-3 (Code Structure Measurement of SPL I)

Course Code:SE3209

Course title: Software Metrics

03/05/2023

Team Members:

1.	Ikra chowdhury Nowkshi	BFH1925019F
2.	Ishrat Jahan Rintu	BFH1925002F
3.	Roichuddin Rana	ASH1925003M
4.	Dhruba Kanti Bakshi	ASH1825018M

Noakhali Science and Technology University (NSTU)

1. Measuring Coupling in Object-Oriented System

Code structure measurement is the process of quantifying the quality of a software's structure. It involves using various metrics to evaluate the design and organization of the code. The goal is to identify potential issues and areas for improvement, as well as to track the progress of development.

The metrics uses to measure the code structure of the SPL1 project:

- Control Flow Diagrams
- Number of decision points
- Cyclomatic Complexity

1.1 Control Flow Diagrams

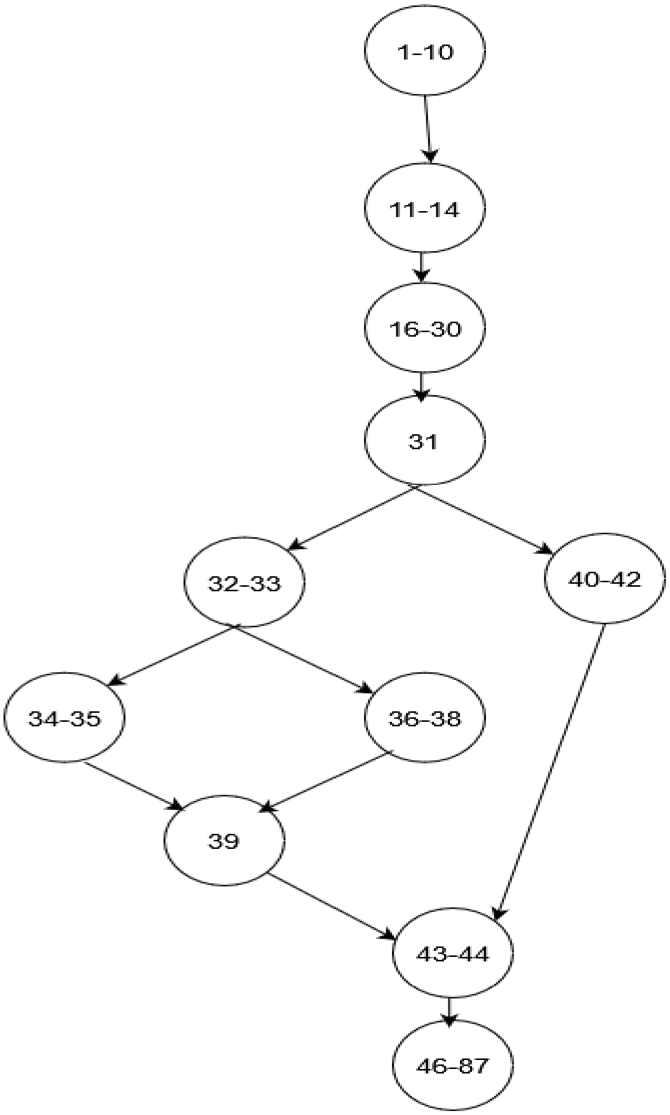
A Control Flow Diagram is a graphical representation of the control flow of a program. It depicts the sequence of operations and the paths of data through a program. The diagram can be used to identify the complexity of the program, and to detect errors in the control flow.

Measurement process:

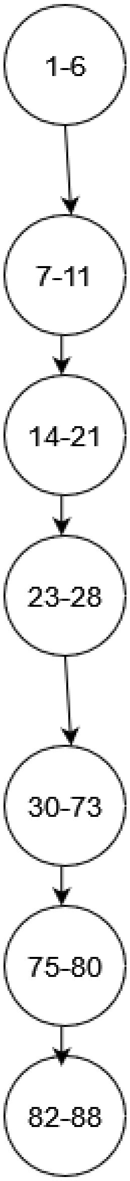
Control Flow Diagrams can be created using a variety of techniques, such as structured programming, flowcharts, and data flow diagrams. The diagram should include all possible paths through the program, and should clearly indicate the sequence of operations and the flow of data.

The control flow diagrams obtained from analyzing the source code of our SPL1 project are given below:

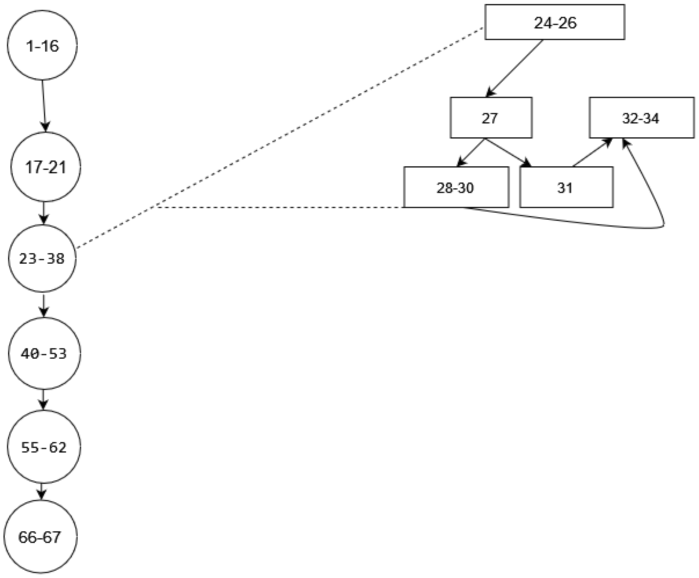
Class: WordSearch

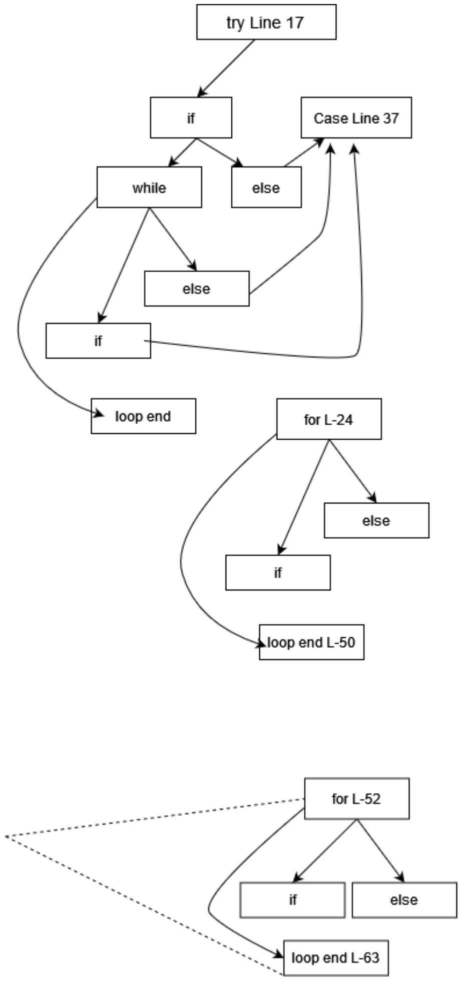


Class: userstructure

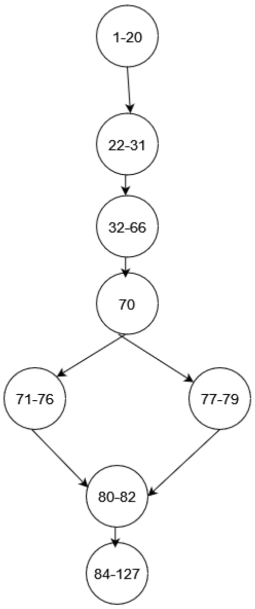


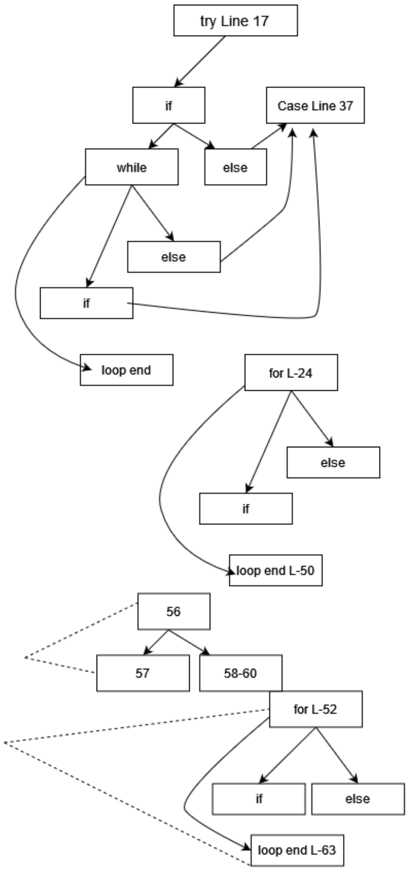
Package: Account->UserData
Class: `objectread`



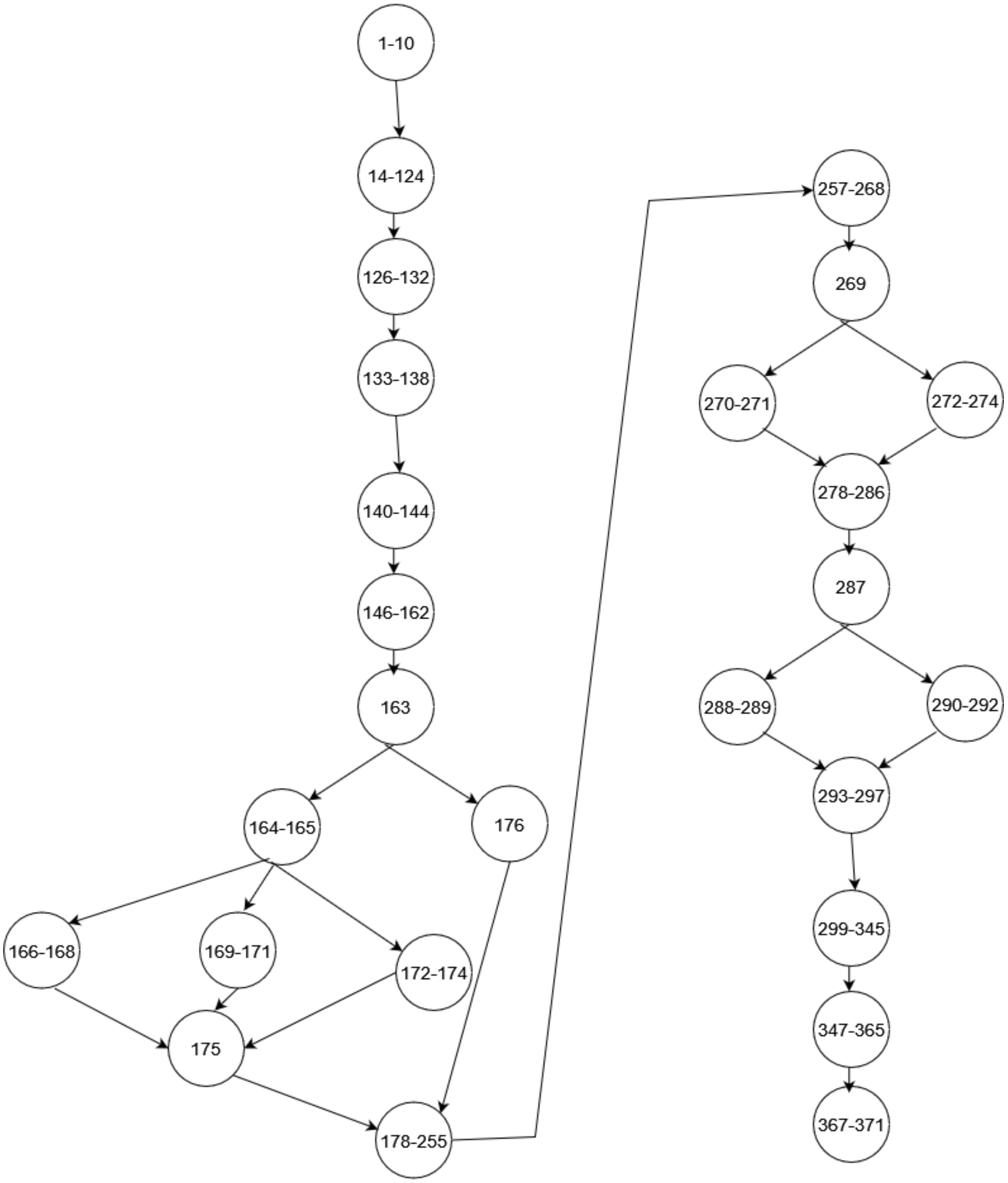


Package: Account->UserData
Class: `ImplementWord`

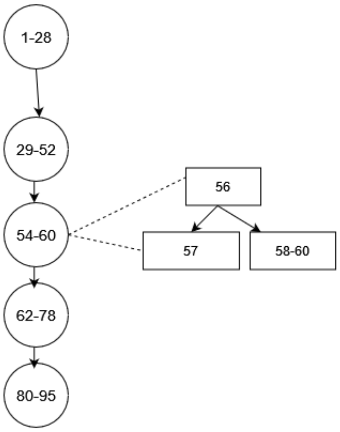


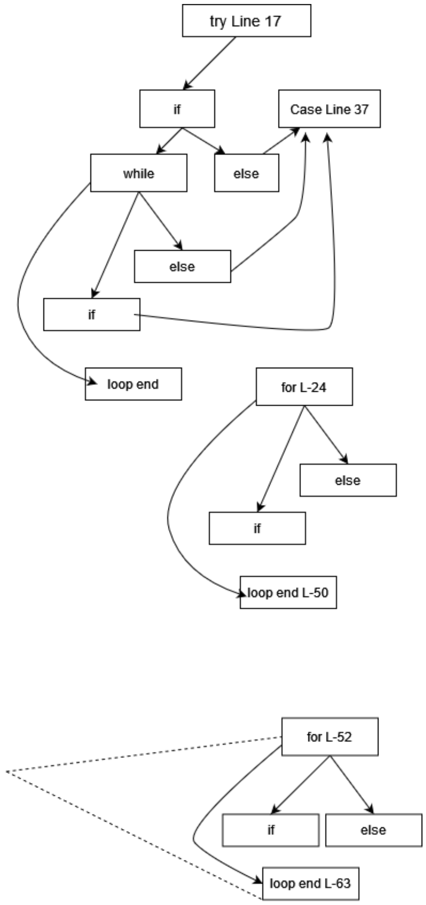


Class: DictionaryMainFrame

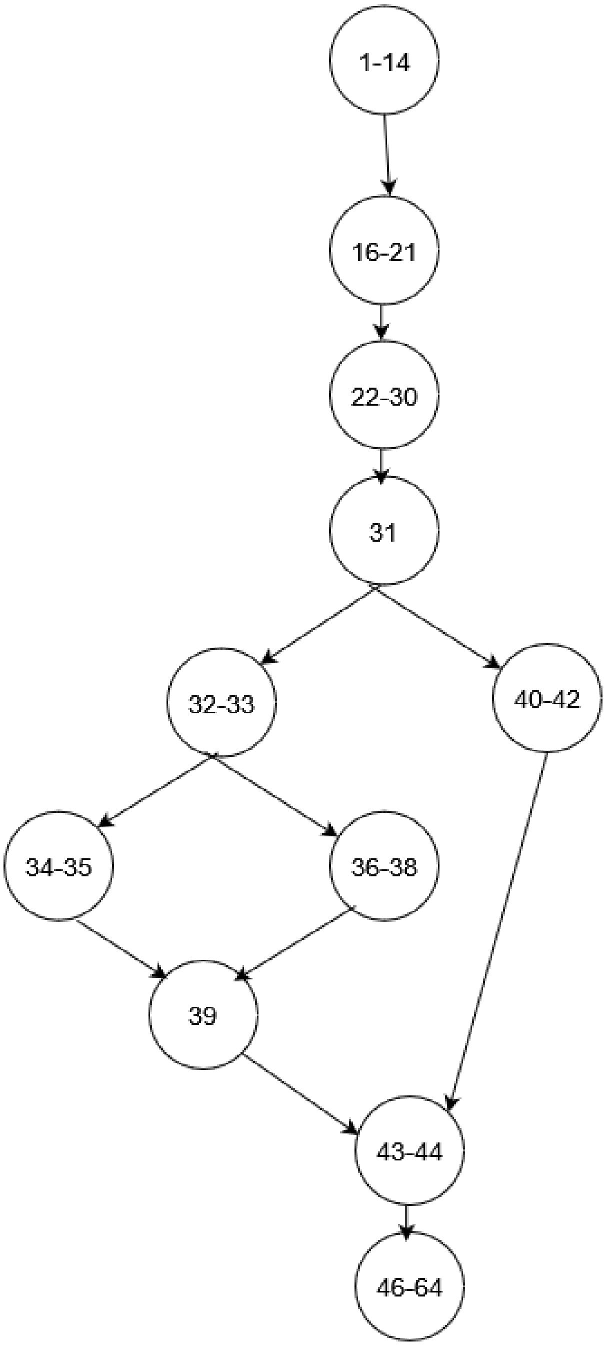


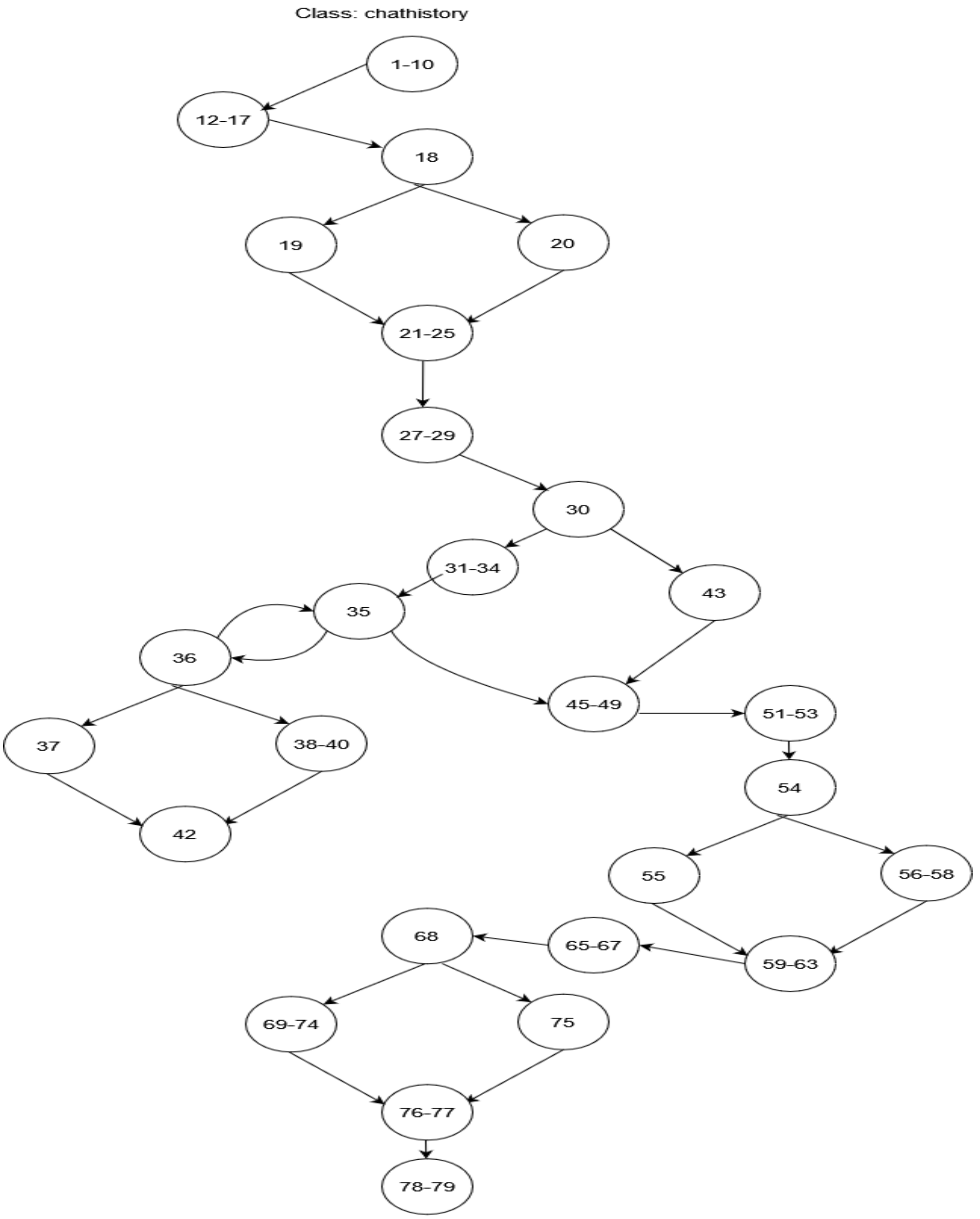
Package: Account->UserData
Class: `Dictionary`



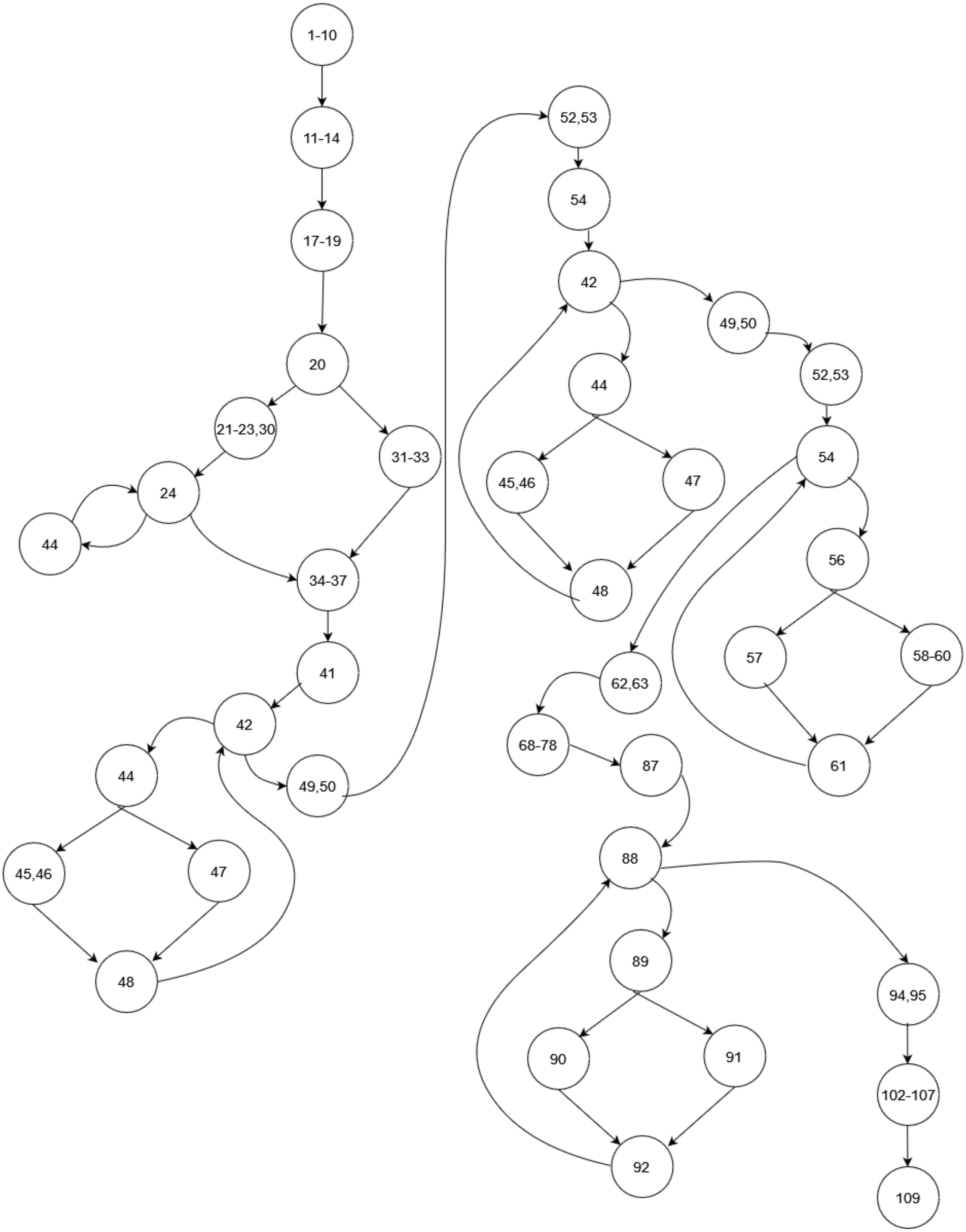


Class: DeleteListedWord

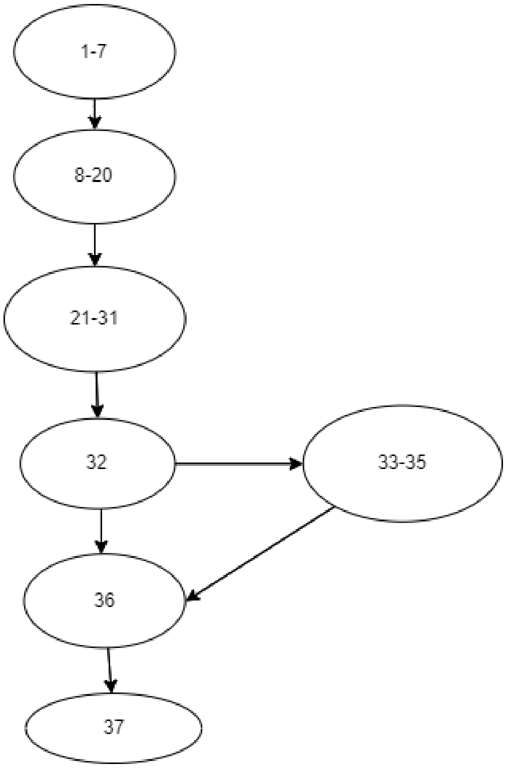



Class: chathistory
1-10
12-17
18
19
20
21-25
27-29
30
31-34
43
35
36
45-49
51-53
37
38-40
42
54
55
56-58
68
65-67
59-63
69-74
75
76-77
78-79

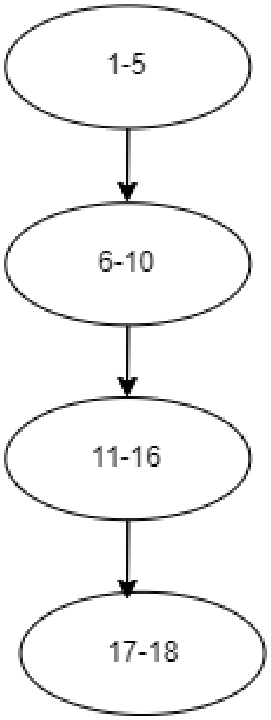
Class: accountdata



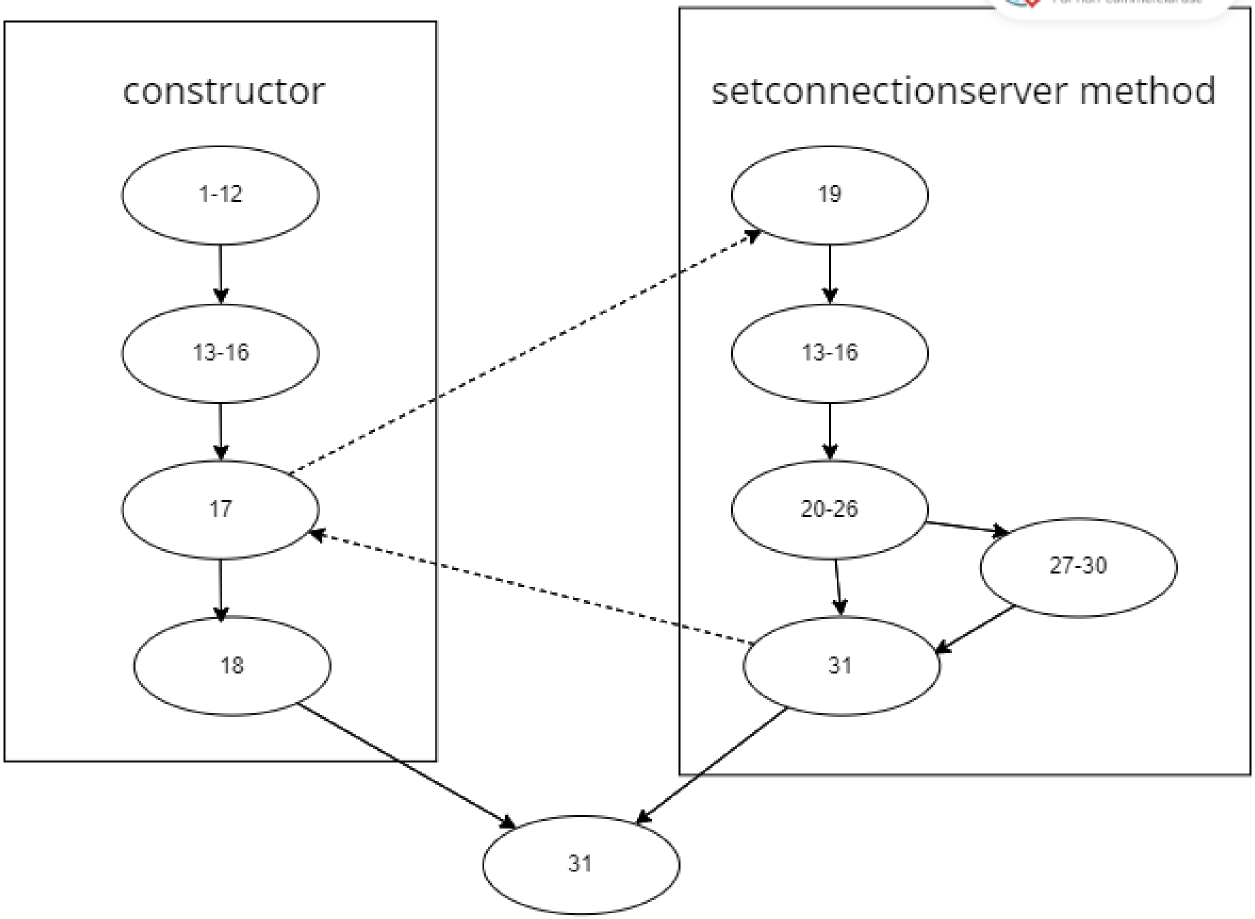
Clintsocket.java



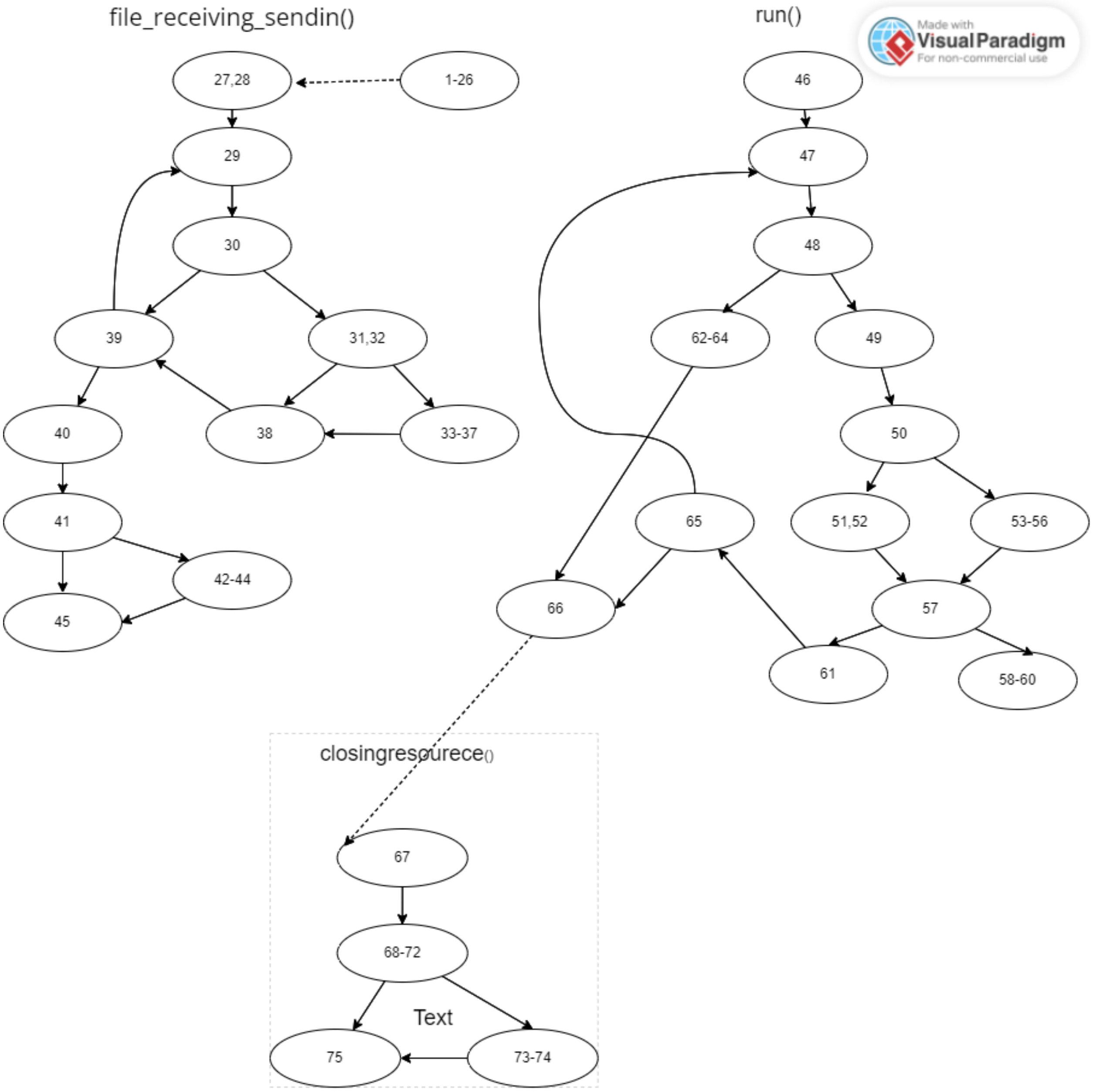
clinthread.java



file_name_transfer.java

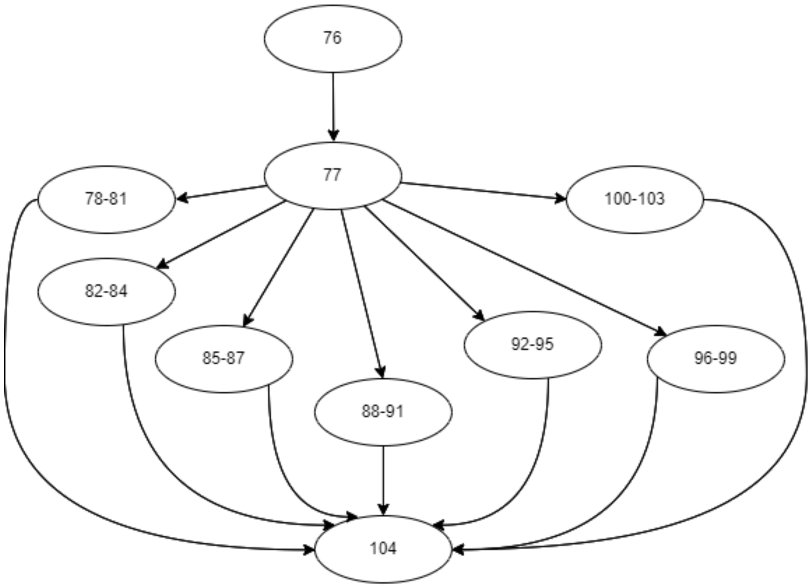


Clinthandaler2.java

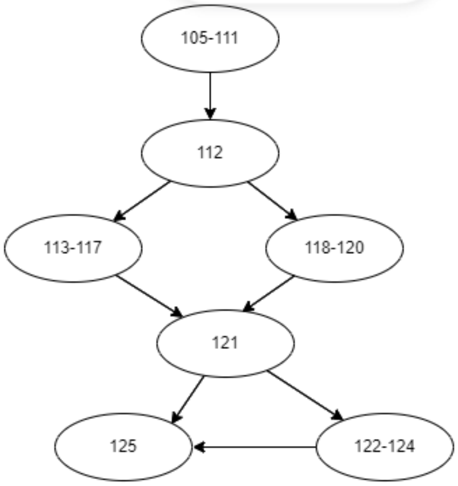


Clinthandaler2.java

allfunction()

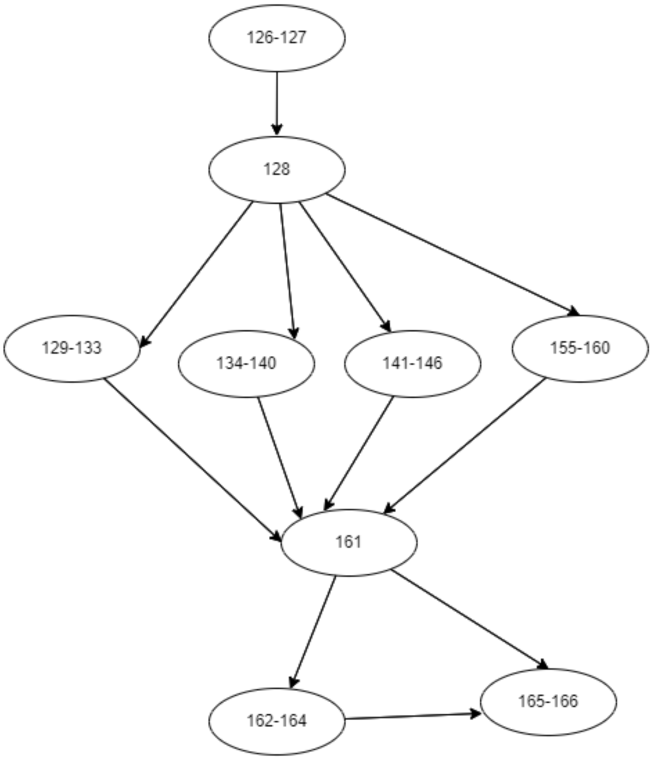


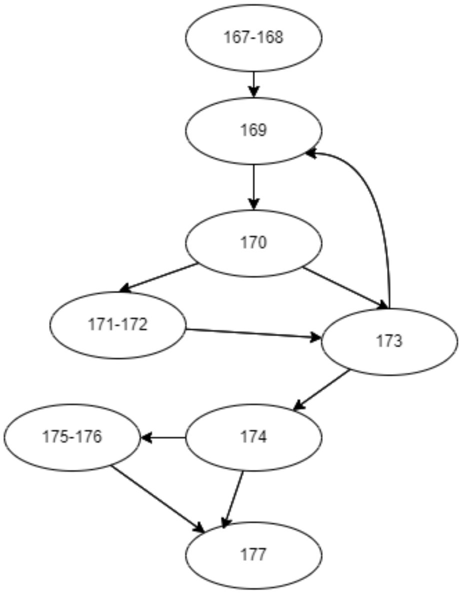
Createregist



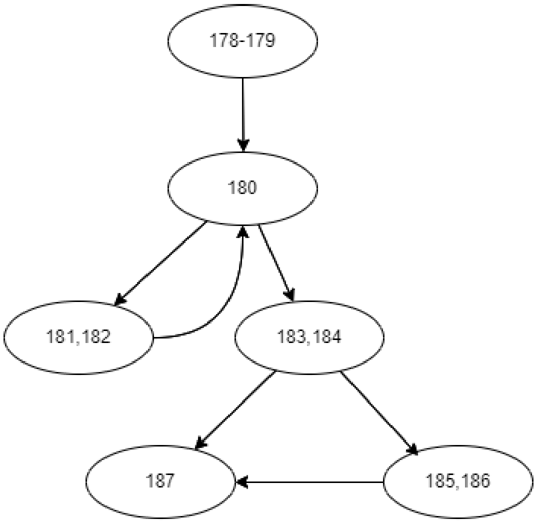
Clinthandaler2.java

msg_to_actualmsg()

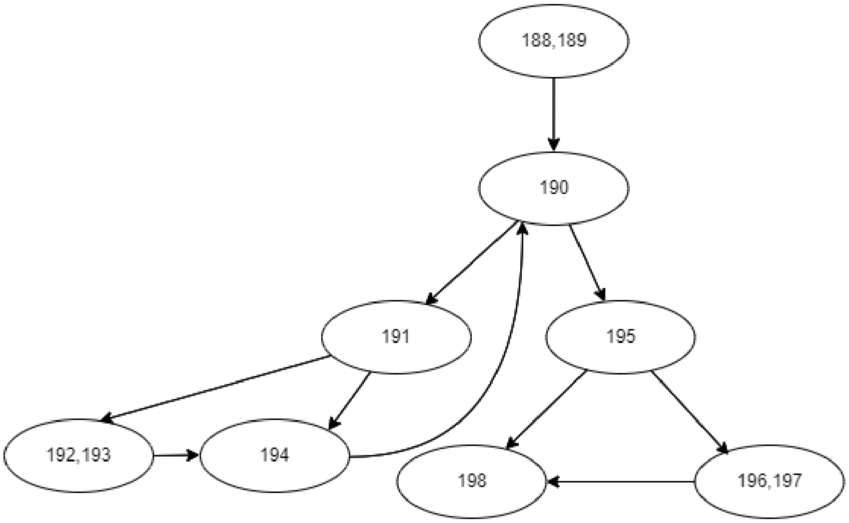




alreadyconnectedpeoplelist()

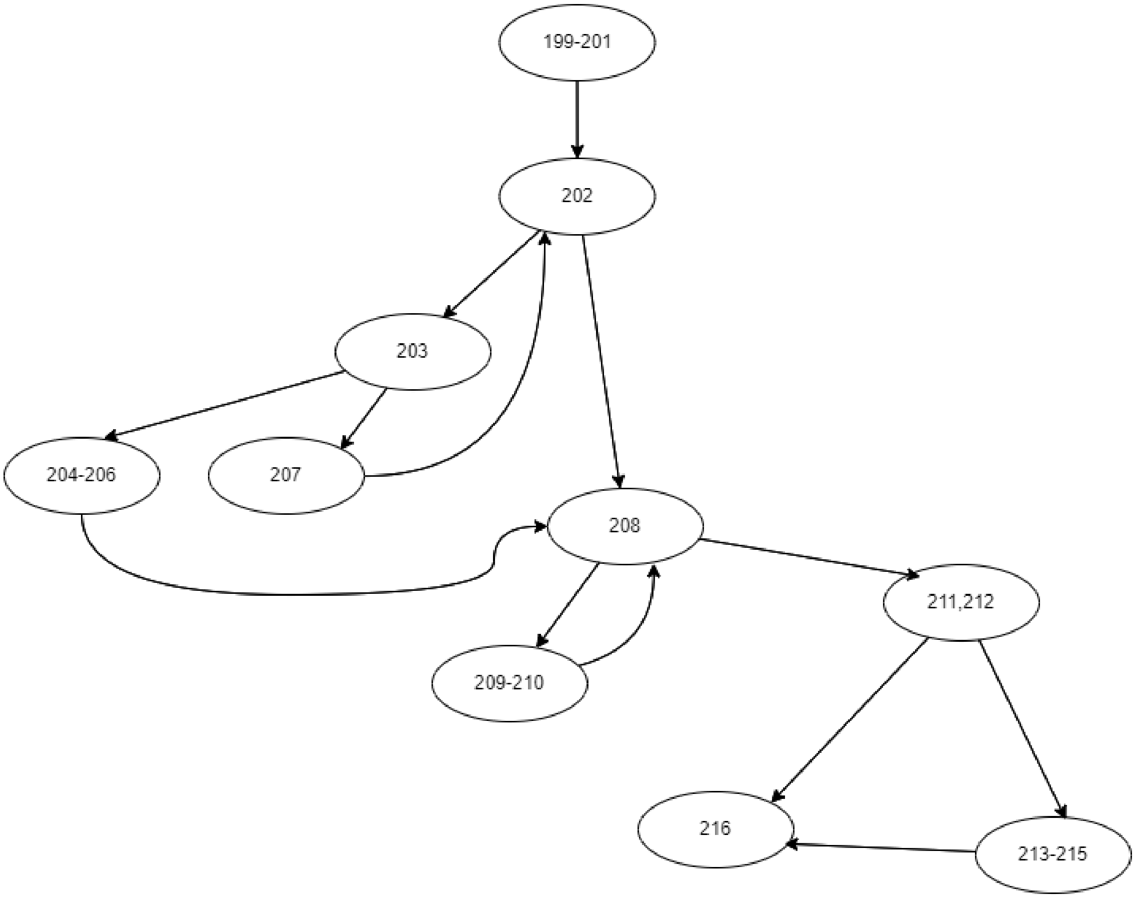


sendtoclint()

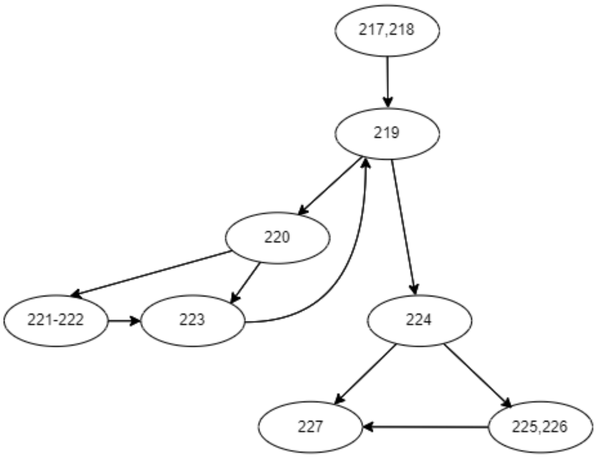


Clinthandaler2.java

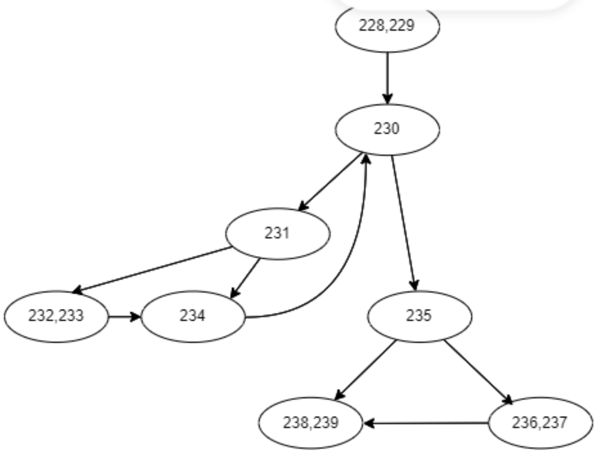
loggedout()



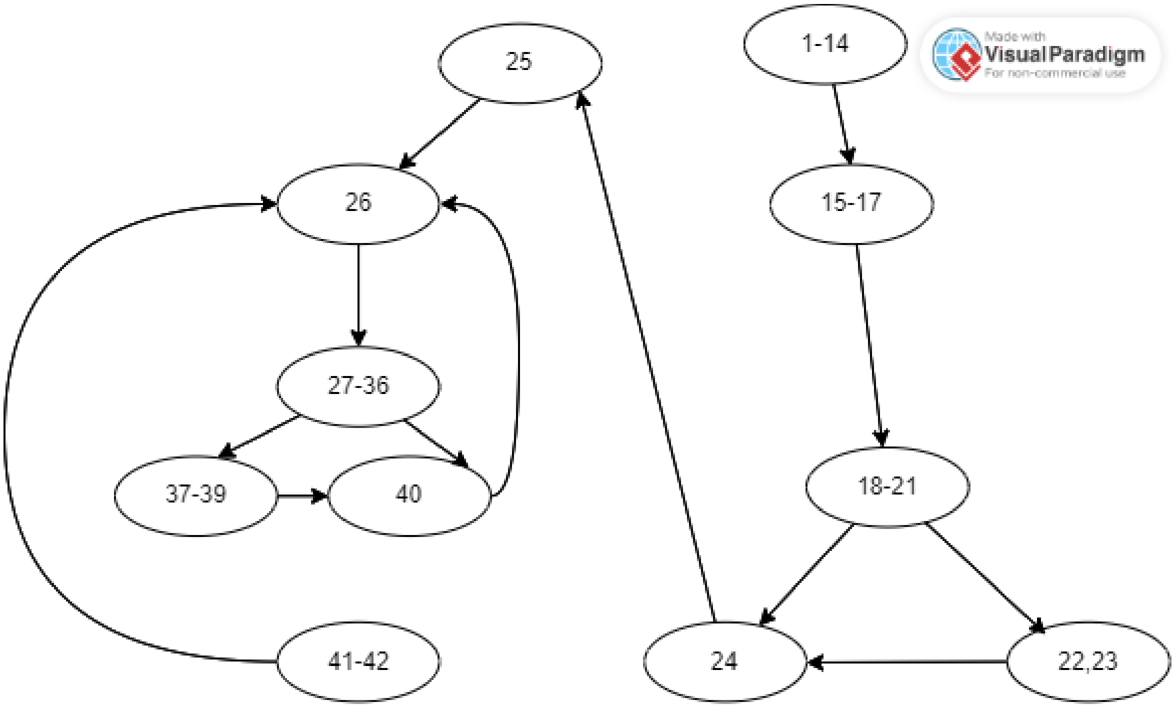
send_friendrequest()



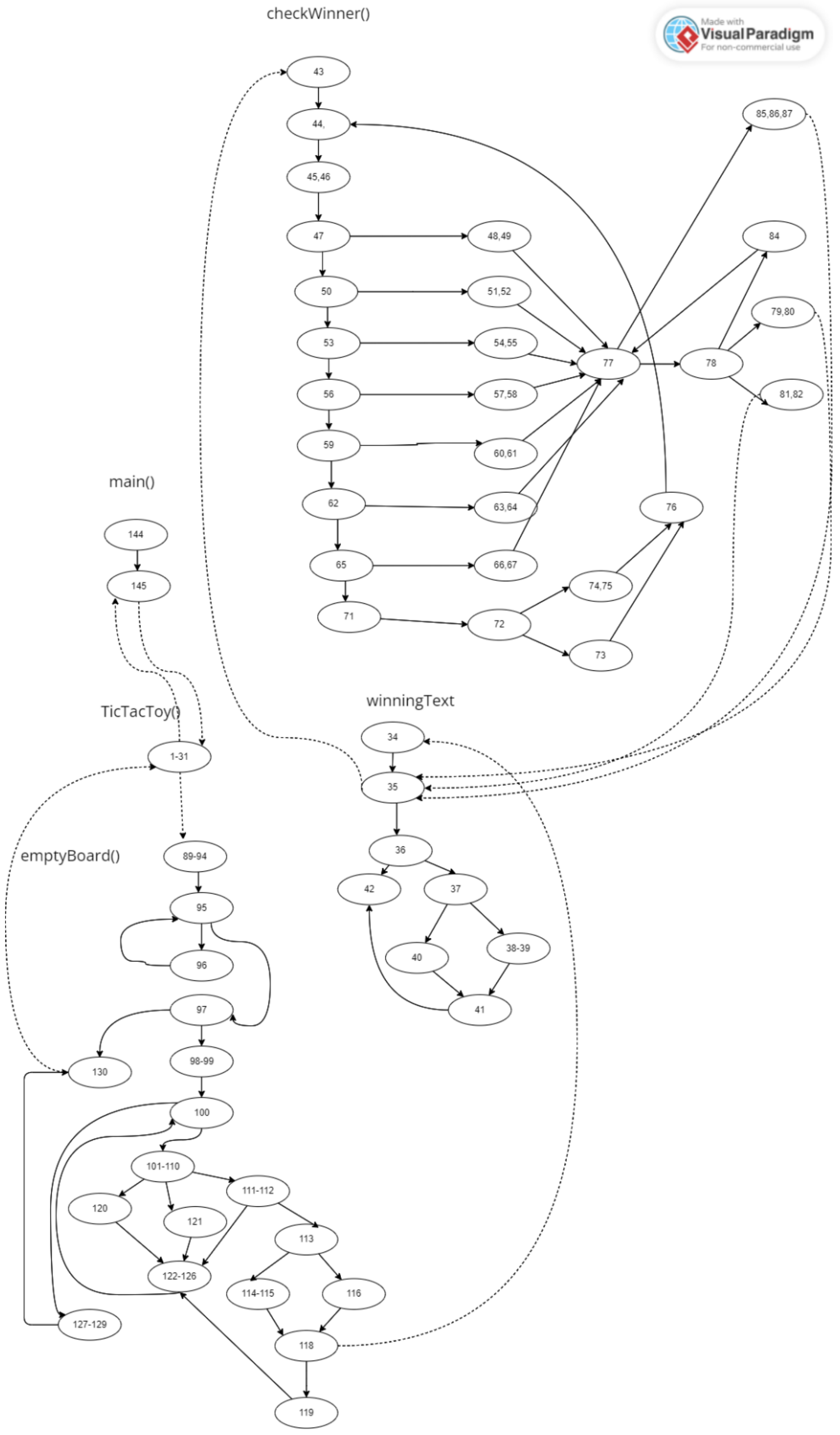
friendrequest_ans(

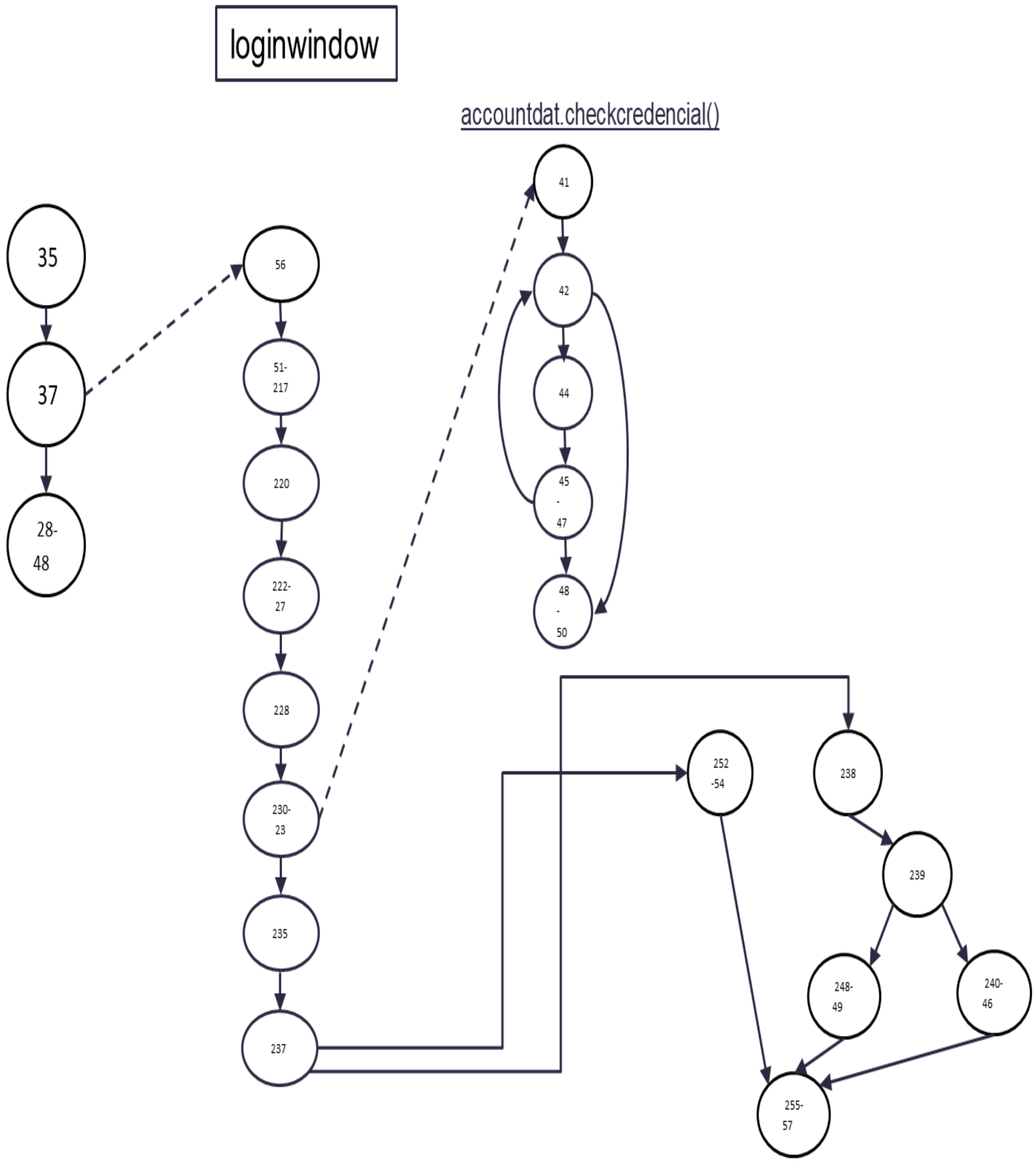


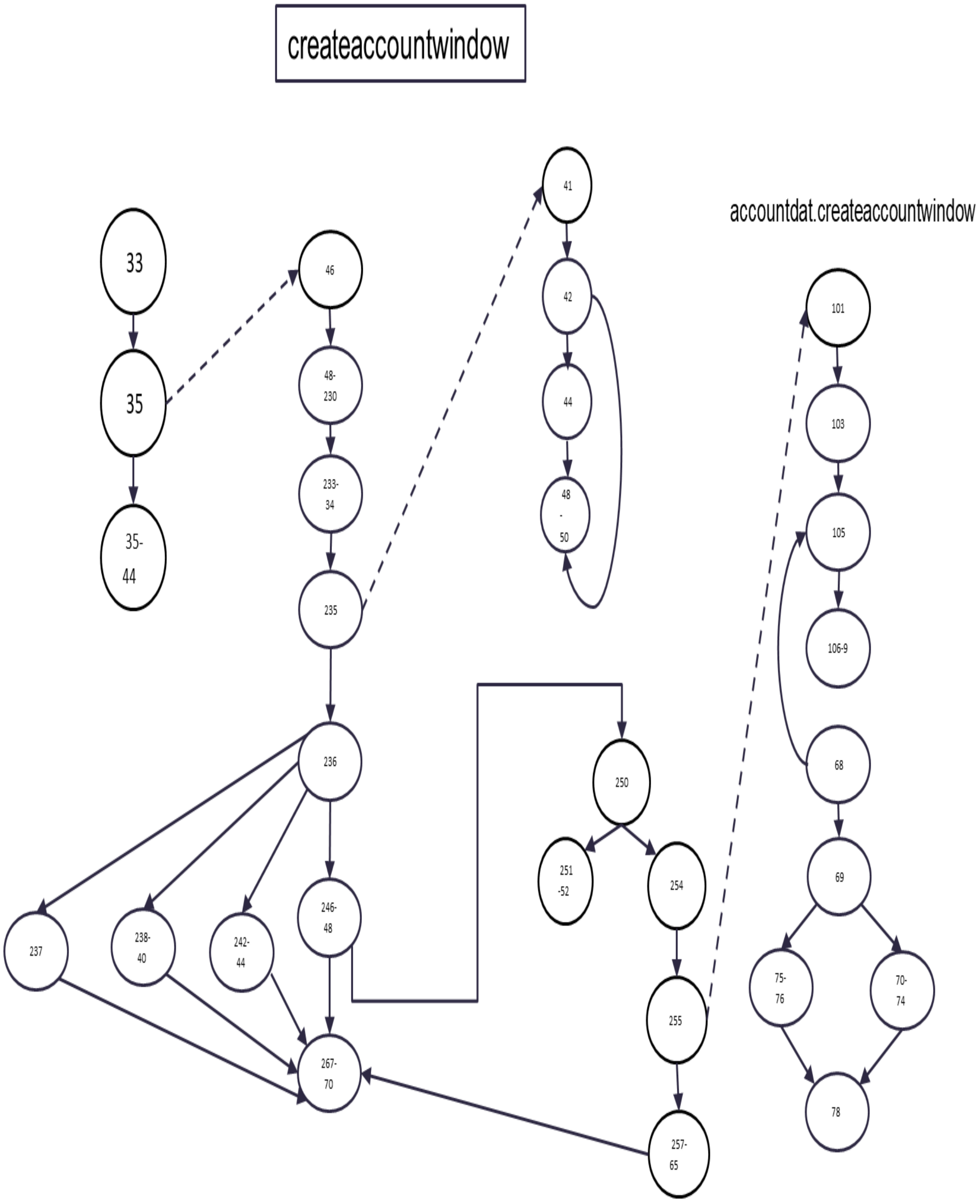
serversocket.java

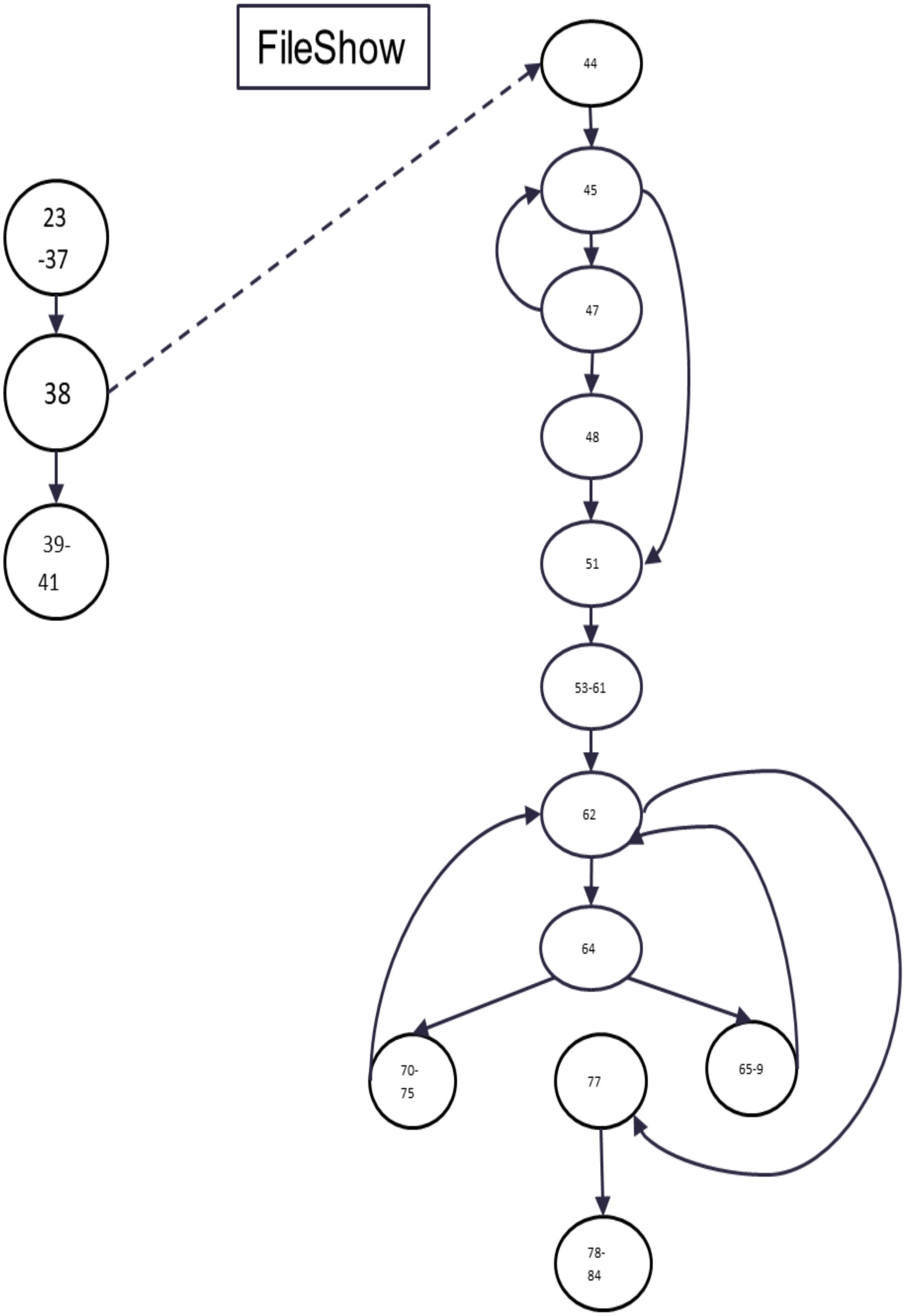


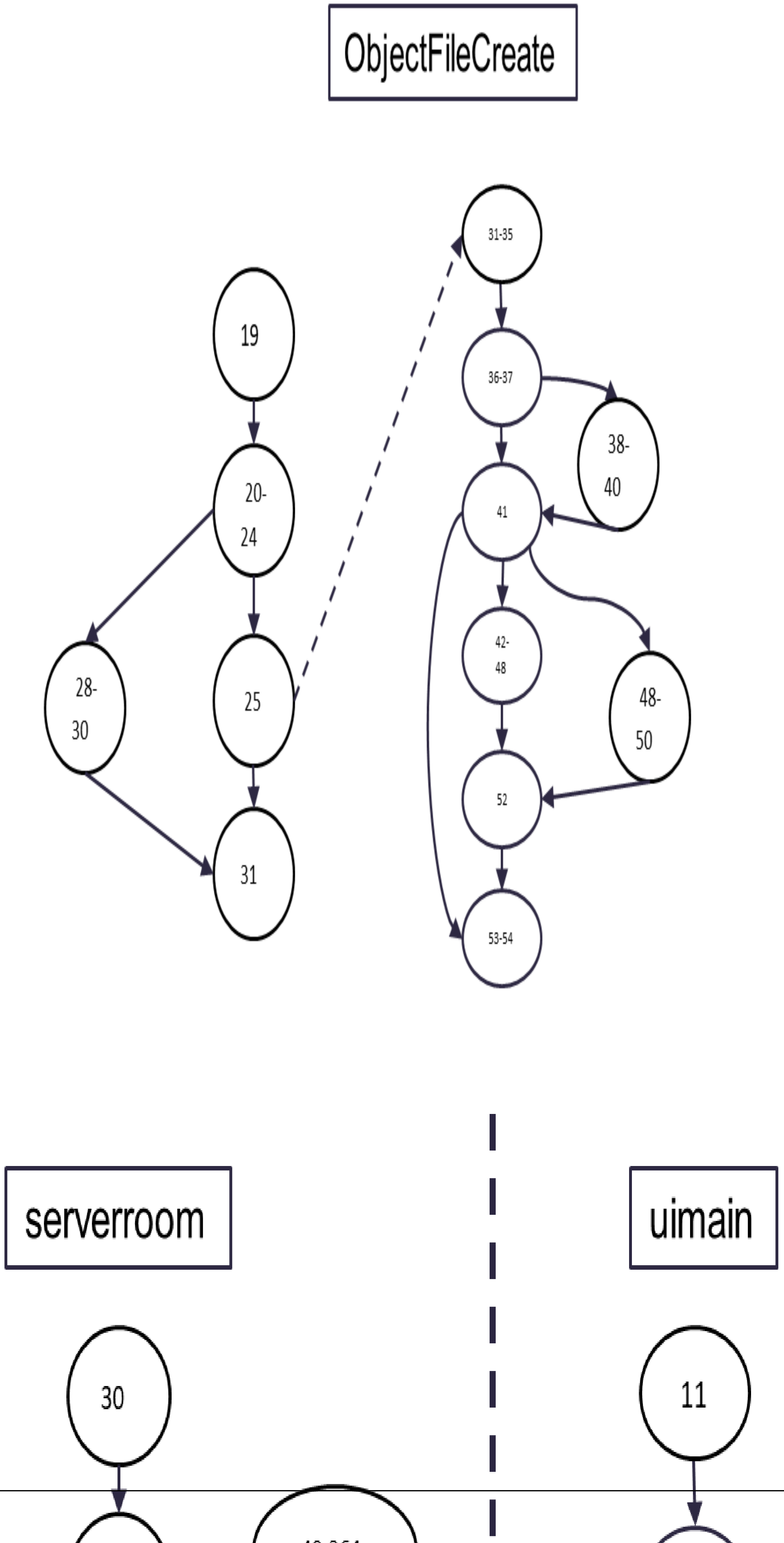
tictactoe.java

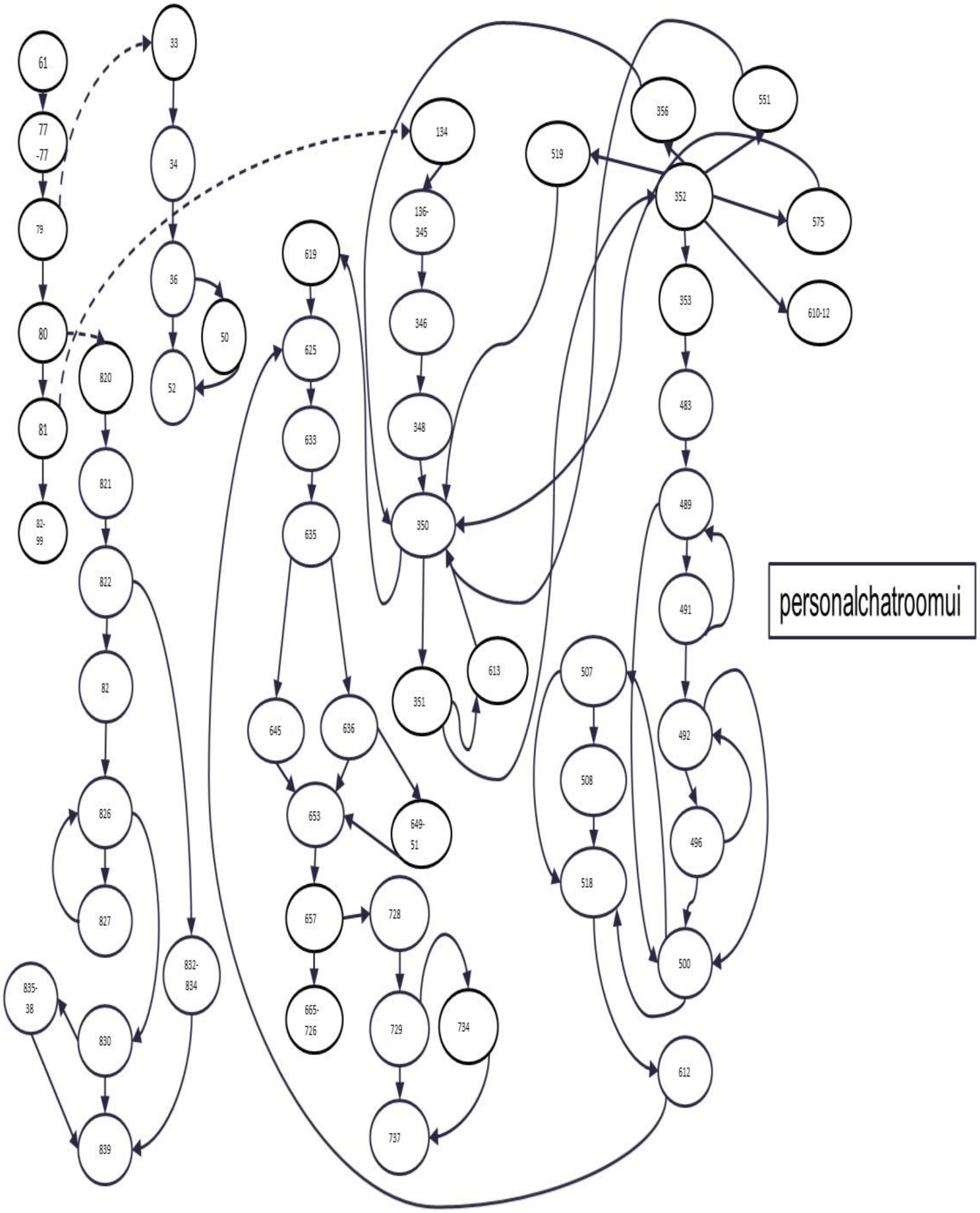



loginwindow
accountdat.checkcredencial()
35
37
28-
48
56
51-
217
220
222-
27
228
230-
23
235
237
41
42
44
45
-
47
48
-
50
252
-54
238
239
248-
49
240-
46
255-
57


createaccountwindow
33
35
35-44
46
48-230
233-34
235
41
42
44
48-50
accountdat.createaccountwindow
101
103
105
106-9
68
69
75-76
70-74
78
236
237
238-40
242-44
246-48
267-70
250
251-52
254
255
257-65


FileShow
23
-37
38
39-
41
44
45
47
48
51
53-61
62
64
70-
75
77
65-9
78-
84


ObjectFileCreate
19
20-24
28-30
25
31
31-35
36-37
38-40
41
42-48
48-50
52
53-54
serverroom
30
uimain
11


61
77
-77
79
80
81
82-
99
33
34
36
50
52
820
821
822
82
826
827
830
835-
38
832-
834
839
619
625
633
635
645
636
653
649-
51
657
665-
726
728
729
734
737
134
136-
345
346
348
350
351
613
519
356
551
352
575
353
610-12
483
489
491
492
496
500
507
508
518
612
personalchatroomui

1.2 Number of decision points

The number of decision points is a metric that indicates the number of points in a program where a decision is made based on a condition. These points typically include if statements, switch statements, and loops with conditions.

Measurement approach:

The process of measuring the number of decision points in a program involves analyzing the program's control flow and identifying the locations where decisions are made based on conditions.

The number of decision points found by analyzing the source code of our SPL1 project is given below:

Class	Number of decision points
userstructure	0
accountdata	10
chathistory	5
objectread	4
FileShow	4
ObjectFileCreate	3
createaccountWindow	6
loginwindow	4
personalchatroomui	24
serverroom	1
TicTacToy	16
clintsocket	1
clintthread	0
file_name_transfer	1

clinthandaler2	31
serversocket	2
DeleteListedWord	2
Dictionary	1
DictionaryMainFrame	4
ImplementWord	0
WordSearch	1

1.3 Cyclomatic complexity

Cyclomatic Complexity is a measure of the number of independent paths through a program. It is calculated using the Control Flow Graph of the program, which represents the control flow of the program in a graphical form. Cyclomatic Complexity can be used to measure the complexity of a program, and to identify areas of the code that are more difficult to understand

Measurement approach:

The process of measuring Cyclomatic Complexity involves creating a Control Flow Graph and using it to calculate the Cyclomatic Complexity of the program. The metric can be obtained using the following formula:

$M = E - N + 2P$

where:

M is the Cyclomatic Complexity

E is the number of edges in the Control Flow Graph

N is the number of nodes in the Control Flow Graph

P is the number of connected components in the Control Flow Graph

Below is the count of Cyclomatic complexity that was obtained by analyzing the source code of our SPL1 project.

Class	Cyclomatic complexity
userstructure	1
accountdata	11
chathistory	6
objectread	5
FileShow	5
ObjectFileCreate	4
createaccountWindow	7
loginwindow	5
personalchatroomui	25
serverroom	2
TicTacToy	17
clintsocket	2
clintthread	1
file_name_transfer	2
clinthandaler2	32
serversocket	3
DeleteListedWord	3
Dictionary	2
DictionaryMainFrame	5
ImplementWord	1
WordSearch	2