Software Testing

Assignment

Software Testing Assignment No: 1

Submitted By:

Muhammad Bilal	22F-3845
Muhammad Ahmed	22F-8814

Submitted To:

Sir Usman Ghous

February 2026

Contents

1 Introduction

This assignment presents a comprehensive analysis of three distinct features through control flow graph construction, cyclomatic complexity calculation, and test path identification using set theory. The features analyzed include Pagination Logic (dividing file content into fixed-size pages), Search & Replace Word (keyword searching with prefix extraction), and Auto-Save Logic (automatic file saving with word count validation).

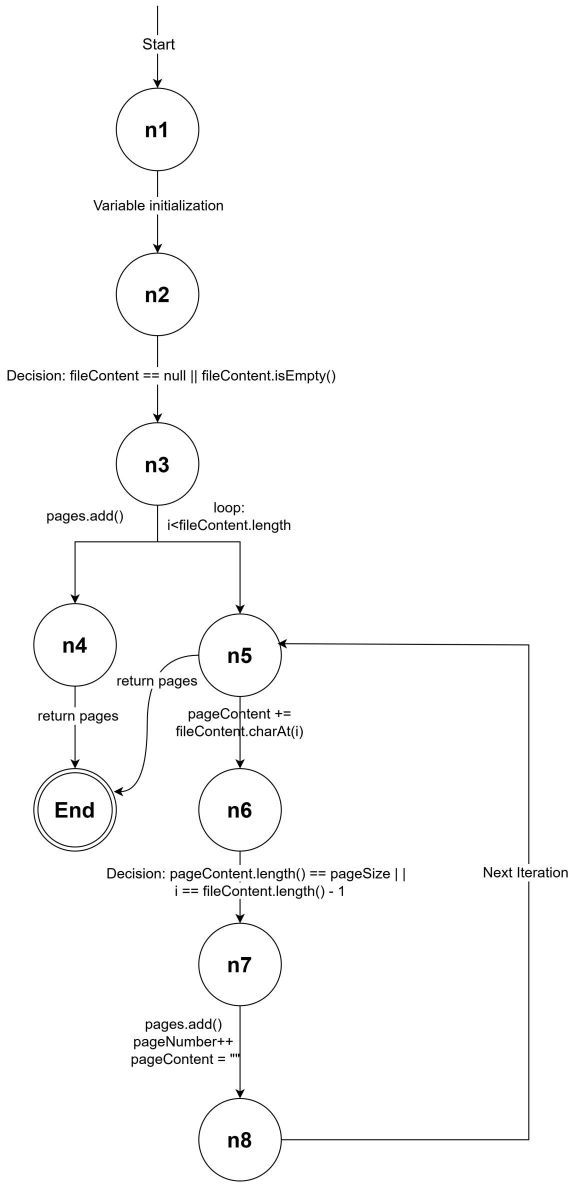
2 Pagination Logic Feature

2.1 Control Flow Graph (CFG)

2.1.1 Step 1: Identify Basic Blocks and Decision Points

Node	Description (Basic Block)
N1	Method entry and initialization: pageSize=100, pageNumber=1, pageContent="", pages = new ArrayList()
N2	Decision: if (fileContent == null || fileContent.isEmpty())
N3	True branch: pages.add(new Pages(...)); return pages;
N4	Loop header: initialize i=0; decision i < fileContent.length()
N5	pageContent += fileContent.charAt(i)
N6	Decision: if (pageContent.length() == pageSize || i == fileContent.length() - 1)
N7	pages.add(new Pages(...)); pageNumber++; pageContent = ""
N8	Loop increment: i++ and back to loop condition (N4)
N9	Method exit: return pages (after loop completion)

2.1.2 Step 2: CFG Diagram


Start
n1
Variable initialization
n2
Decision: fileContent == null || fileContent.isEmpty()
n3
pages.add()
loop: i<fileContent.length
n4
n5
return pages
return pages
End
pageContent += fileContent.charAt(i)
n6
Decision: pageContent.length() == pageSize | | i == fileContent.length() - 1
Next Iteration
n7
pages.add() pageNumber++ pageContent = ""
n8

2.1.3 Step 3: Edge List

Edge	From → To	Condition
e1	N1 → N2	Method entry to null/empty check
e2	N2 → N3	fileContent == null || fileContent.isEmpty() is true
e3	N2 → N4	fileContent == null || fileContent.isEmpty() is false
e4	N4 → N5	Loop condition i < fileContent.length() is true
e5	N4 → N9	Loop condition i < fileContent.length() is false (loop exit)
e6	N5 → N6	After appending character to pageContent
e7	N6 → N7	pageContent.length() == pageSize || i == fileContent.length() - 1 is true
e8	N6 → N8	pageContent.length() == pageSize || i == fileContent.length() - 1 is false
e9	N7 → N8	After adding page and resetting pageContent
e10	N8 → N4	Loop increment and next iteration (i++)

2.2 Cyclomatic Complexity Calculation

Using the formula:

$$V(G) = E - N + 2P \tag{1}$$

Where:

- E (Edges) = 10
- N (Nodes) = 9
- P (Connected components) = 1

$$V(G) = 10 - 9 + 2(1) = 3 \tag{2}$$

Verification using decision count:
Decision nodes in the CFG:

- N2: null or empty check
- N4: loop condition
- N6: page size or last character check

Although three predicate nodes appear in the source code, the loop structure is represented as a single cycle in the CFG. Therefore, the number of **linearly independent paths** for this graph is:

$$\boxed{V(G) = 3}$$

This means that at least **three independent test paths** are required for complete basis path coverage.

2.3 Test Paths (Set Theory)

2.3.1 Independent Path Set

The set of independent paths through the CFG is defined as:

$$P = \{p_1, p_2, p_3\} \tag{3}$$

Path 1 (Empty input):

$$p_1 = \langle N_1, N_2, N_3 \rangle$$

Path 2 (Single short page):

$$p_2 = \langle N_1, N_2, N_4, N_5, N_6, N_7, N_8, N_4, N_9 \rangle$$

Path 3 (Multiple pages):

$$p_3 = \langle N_1, N_2, N_4, N_5, N_6, N_8, N_4, N_5, N_6, N_7, N_8, N_4, N_9 \rangle$$

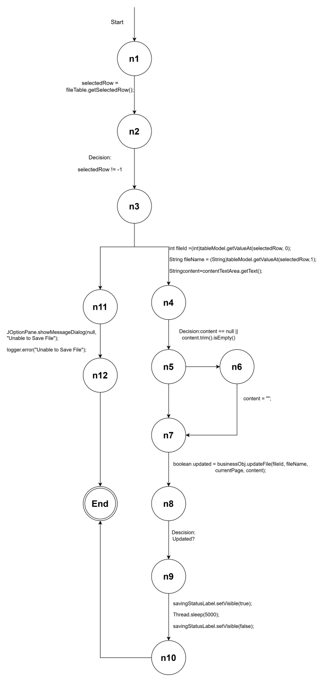
3 Auto-Save Feature

3.1 Control Flow Graph (CFG)

3.1.1 Step 1: Identify Basic Blocks and Decision Points

Node	Description (Basic Block / Decision)
N1	Method entry
N2	selectedRow = fileTable.getSelectedRow()
N3	Decision: if (selectedRow != -1)
N4	Retrieve fileId, fileName, and content from table and text area
N5	Decision: if (content == null || content.trim().isEmpty())
N6	content = ””
N7	updated = businessObj.updateFile(fileId, fileName, currentPage, content)
N8	Decision: if (updated)
N9	savingStatusLabel.setVisible(true); Thread.sleep(5000); savingStatusLabel.setVisible(false)
N10	Method exit (normal return after update path)
N11	Show error dialog: JOptionPane.showMessageDialog(null, ”Unable to Save File”)
N12	logger.error(”Unable to Save File”)
N13	Method exit (error path return)

3.1.2 Step 2: CFG Diagram


Start
n1
selectedRow = fileTable.getSelectedRow();
n2
Decision:
selectedRow != -1
n3
int fileId =(int)tableModel.getValueAt(selectedRow, 0);
String fileName = (String)tableModel.getValueAt(selectedRow,1);
Stringcontent=contentTextArea.getText();
n11
JOptionPane.showMessageDialog(null, "Unable to Save File");
logger.error("Unable to Save File");
n12
n4
Decision:content == null || content.trim().isEmpty()
n5
n6
content = "";
n7
boolean updated = businessObj.updateFile(fileId, fileName, currentPage, content);
End
n8
Descision: Updated?
n9
savingStatusLabel.setVisible(true);
Thread.sleep(5000);
savingStatusLabel.setVisible(false);
n10

3.1.3 Step 3: Edge List

Edge	From → To	Condition
e1	N1 → N2	Method entry
e2	N2 → N3	After retrieving selectedRow
e3	N3 → N4	selectedRow ≠ -1 (true branch)
e4	N3 → N11	selectedRow = -1 (false branch)
e5	N11 → N12	Show error dialog
e6	N12 → N13	Log error and exit (error path)
e7	N4 → N5	After retrieving fileId, fileName, content
e8	N5 → N6	content == null || content.trim().isEmpty() is true
e9	N5 → N7	content == null || content.trim().isEmpty() is false
e10	N6 → N7	After setting content to empty string
e11	N7 → N8	After calling updateFile(...)
e12	N8 → N9	updated == true
e13	N8 → N13	updated == false (exit without showing status)
e14	N9 → N10	After showing and hiding saving status label
e15	N10 → N13	Normal method exit

Total Edges: E = 15

3.2 Cyclomatic Complexity Calculation

3.2.1 Node Count

N = 13 nodes (N1 through N13)

3.2.2 Edge Count

E = 15 edges (e1 through e15)

3.2.3 Connected Components

P = 1 (single method, single connected graph)

3.2.4 Formula

$$V(G) = E - N + 2P \tag{4}$$

$$V(G) = 15 - 13 + 2(1) = 4 \tag{5}$$

3.2.5 Verification using Decision Count

Decision nodes in the CFG:

- N3: selectedRow $\neq$ -1 check
- N5: content null/empty check
- N8: updated success check

Total decisions = 3

$$V(G) = \text{decisions} + 1 = 3 + 1 = 4 \tag{6}$$

Cyclomatic Complexity:

$$\boxed{V(G) = 4}$$

This indicates that at least **four independent test paths** are required to achieve complete basis path coverage.

3.3 Test Paths (Set Theory)

3.3.1 Independent Path Set

The set of all linearly independent paths:

$$P = \{p_1, p_2, p_3, p_4\} \tag{7}$$

Path 1 (No Row Selected - Error Path):

$$p_1 = \langle N_1, N_2, N_3, N_{11}, N_{12}, N_{13} \rangle \tag{8}$$

Description: No row selected, error dialog shown and error logged.
Input: selectedRow = -1
Expected: Error dialog "Unable to Save File", logger.error called

Path 2 (Empty Content, Update Fails):

$$p_2 = \langle N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_{13} \rangle \tag{9}$$

Description: Row selected, content empty/null, updateFile returns false.
Input: selectedRow = 0, content = ""
Expected: No saving status shown, method exits normally

Path 3 (Non-Empty Content, Update Fails):

$$p_3 = \langle N_1, N_2, N_3, N_4, N_5, N_7, N_8, N_{13} \rangle \tag{10}$$

Description: Row selected, content non-empty, updateFile returns false.
Input: selectedRow = 0, content = "Some text"
Expected: No saving status shown, method exits normally

Path 4 (Successful Auto-Save):

$$p_4 = \langle N_1, N_2, N_3, N_4, N_5, N_7, N_8, N_9, N_{10}, N_{13} \rangle \tag{11}$$

Description: Row selected, update succeeds, saving status is displayed and hidden.
Input: selectedRow = 0, content = "Valid content", updateFile returns true
Expected: Saving status label visible for 5 seconds, then hidden

3.4 Complexity Comparison

Feature	Nodes (N)	Edges (E)	V(G)
Pagination Logic	9	10	3
Auto-Save	13	15	4