

## Test Case for Use Case 2

```
myTeam.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Create the custom dialog
        JDialog dialog = new JDialog((Frame)null, "Enter Project ID",
true);
        dialog.setLayout(new FlowLayout());

        JLabel label = new JLabel("Enter the Project ID you want to
see your team for:");
        JTextField textField = new JTextField(20);
        JButton okButton = new JButton("OK");
        JButton findTeamButton = new JButton("Find Team");
        JButton cancelButton = new JButton("Cancel");

        dialog.add(label);

        dialog.add(textField);
        dialog.add(okButton);
        dialog.add(cancelButton);
        dialog.add(findTeamButton);

        // Handle OK button action
        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                String projectId = textField.getText();

                if (!projectId.trim().isEmpty() &&
\\4\\
isValidProjectId(projectId.trim())) {

                    if (userExistsInTeamProjects(user.getUsername(),
projectId.trim())) {
                        JOptionPane.showMessageDialog(dialog, "You
have a team, moving to CoShare!");
                        dialog.dispose();

                        dispose();
                        new StudentTeam(null, user, id,
projectId.trim());
                    } else {
                        dialog.dispose();

                        dispose();
                        new studentFindsTeam(null, user,
projectId.trim());
                    }
                } else {
                    JOptionPane.showMessageDialog(dialog, "Invalid
```

```

Project ID. Please try again.");
        }
    });
}

```

8

\5\

```

public boolean userExistsInTeamProjects(String username, String
projectId) {
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String user = "root";
    String password = "W45@jqr#8CX";
    String query = "SELECT COUNT(*) AS count FROM teams_projects
WHERE username = ? AND id_team_project = ?";

```

1

```

Connection conn = null;
PreparedStatement stmt = null;
ResultSet rs = null;

```

```

try {
    conn = DriverManager.getConnection(url, user, password);

```

```

    // Prepare and execute the SQL statement
    stmt = conn.prepareStatement(query);

```

```

    stmt.setString(1, username);

```

2

```

    stmt.setString(2, projectId);
    rs = stmt.executeQuery();

```

```

    // Check if the count is greater than zero

```

3

4

```

    if (rs.next() && rs.getInt("count") > 0) {

```

```

        return true;
    }

```

5

```

} catch (Exception ex) {

```

```

    ex.printStackTrace();

```

6

```

} finally {
    // Close resources

```

```

    try {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();

```

```

        if (conn != null) conn.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

7

```

return false; // No records found
}

```

8

\4\

```
public boolean isValidProjectId(String projectId) {
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String user = "root";

    String password = "W45@jqr#8CX"; 1
    String query = "SELECT COUNT(*) AS count FROM projects WHERE
idprojects = ?";

    try (Connection conn = DriverManager.getConnection(url, user,
password); 2
        PreparedStatement stmt = conn.prepareStatement(query)) {

        stmt.setString(1, projectId); 3

        try (ResultSet rs = stmt.executeQuery()) { 4
            if (rs.next() && rs.getInt("count") > 0) { 5 6
                return true; // Project ID exists 7
            }
        } catch (Exception ex) {
            ex.printStackTrace(); 8
        }

        return false; // Project ID does not exist 9
    }
}
```

### Inside StudentTeam class:

```
uploadFilesButton.addActionListener(e -> {

    JFileChooser fileChooser = new JFileChooser(); 1
    int option = fileChooser.showOpenDialog(this);

    if (option == JFileChooser.APPROVE_OPTION) { 2
        selectedFile = fileChooser.getSelectedFile();
        fileManager.setSelectedFile(selectedFile);
        try {
            String fileContent = new 3
        }
    }
}
```

```

String(Files.readAllBytes(selectedFile.toPath()),
StandardCharsets.UTF_8);
    fileName.setText(selectedFile.getName());
} catch (Exception ex) {
    JOptionPane.showMessageDialog(this, "Error reading file:
" + ex.getMessage(), "File Error", JOptionPane.ERROR_MESSAGE);
}
}
});

```

```

submitButton.addActionListener(e -> {
    String branchName = textField4.getText();

    if (!fileName.getText().trim().isEmpty() &&
        !branchName.isEmpty()) {
        try {
            byte[] fileContent =
Files.readAllBytes(selectedFile.toPath());

            String checkResult =
fileManager.checkFileProjects(selectedFile.getName(), fileContent);
            Dialog dialog = new Dialog(StudentTeam.this,
selectedFile, user, teamID);

            if ("sameName".equals(checkResult)) {
                dialog.showFileExistsDialog(branchName, fileContent);

            } else if ("sameContent".equals(checkResult)) {
                dialog.sameContentDialog(branchName, fileContent);
            } else {
                fileManager.insertFileIntoDatabase(branchName,
fileContent);
            }

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(this, "Error: " +
ex.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {

```

```

        JOptionPane.showMessageDialog(this, "Please select a file and
specify a branch.", "Error", JOptionPane.WARNING_MESSAGE);
    }
});

```

\4\:

```

public String checkFileProjects(String fileName, byte[] fileContent)
throws SQLException, NoSuchAlgorithmException {
    String fileHash = hashFileContent(fileContent);
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";

    String password = "W45@jqr#8CX";

    try (Connection conn = DriverManager.getConnection(url, username,
password);
        PreparedStatement stmtName = conn.prepareStatement("SELECT
fileName FROM co_share_file_projects WHERE fileName = ?");
        PreparedStatement stmtContent =
conn.prepareStatement("SELECT fileContent FROM co_share_file_projects
WHERE fileContent = ?")) {

        // Check for same file name

        stmtName.setString(1, fileName);

        ResultSet rsName = stmtName.executeQuery();

        if (rsName.next()) {
            return "sameName";

        }

        // Check for same file content

        stmtContent.setString(1, fileHash);

        ResultSet rsContent = stmtContent.executeQuery();

        if (rsContent.next()) {
            return "sameContent";

        }

        return "allOk";
    }
}

```

\6\:

```

public void updateVersion(String branchName, byte[] fileContent, File
selectedFile, User user){
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";
    String password = "W45@jqr#8CX";
    String selectBranchIdSQL = "SELECT idbranch, teamId FROM
co_share_branch_projects WHERE branchName = ?";
    String insertFile = "INSERT INTO co_share_file_projects
(fileName, fileType, fileSize, dateUploaded, ownerUsername, branchId,
version, description, teamID, fileContent) VALUES (?, ?, ?, ?, ?, ?,
?, ?, ?, ?)";
    String selectLatestVersion = "SELECT version FROM
co_share_file_projects WHERE branchId = ? ORDER BY dateUploaded DESC
LIMIT 1";

    try (Connection conn = DriverManager.getConnection(url, username,
password);
        PreparedStatement selectBranchStmt =
conn.prepareStatement(selectBranchIdSQL);

        PreparedStatement insertStmt =
conn.prepareStatement(insertFile);
        PreparedStatement selectVersionStmt =
conn.prepareStatement(selectLatestVersion)) {

        selectBranchStmt.setString(1, branchName);
        ResultSet rsBranch = selectBranchStmt.executeQuery();

        if (rsBranch.next()) {
            int branchId = rsBranch.getInt("idbranch");
            int teamID = rsBranch.getInt("teamId");

            selectVersionStmt.setInt(1, branchId);
            ResultSet rsVersion = selectVersionStmt.executeQuery();
            String version = "0.1"; // Default version if no records
            found

            if (rsVersion.next()) {
                float lastVersion =
Float.parseFloat(rsVersion.getString("version"));
                version = String.format("%.1f", lastVersion + 0.1);
            }

            insertStmt.setString(1, selectedFile.getName());
            insertStmt.setString(2,
getFileExtension(selectedFile.getName()));
            insertStmt.setLong(3, selectedFile.length());
            insertStmt.setTimestamp(4, new
Timestamp(System.currentTimeMillis()));
            insertStmt.setString(5, user.getUsername());
            insertStmt.setInt(6, branchId);
            insertStmt.setString(7, version);
            insertStmt.setNull(8, Types.VARCHAR);
            insertStmt.setString(9, Integer.toString(teamID));

```

```

        insertStmt.setBytes(10, fileContent);
        6
        insertStmt.executeUpdate();
        JOptionPane.showMessageDialog(null, "New version " +
version + " of the file created successfully.", "Success",
JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(null, "Branch not found.",
"Error", JOptionPane.ERROR_MESSAGE); 7
    }

    } catch (SQLException ex) { 8
        JOptionPane.showMessageDialog(null, "Database error: " +
ex.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

\8\:

```

public void replaceFile(String teamId){
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";

    String password = "W45@jqr#8CX"; 1
    String selectIdSQL = "SELECT idfile FROM co_share_file_projects
WHERE fileName = ? AND teamID = ?";
    String updateFileSQL = "UPDATE co_share_file_projects SET
fileName = ? WHERE idfile = ? AND teamID = ?";

    try (Connection conn = DriverManager.getConnection(url, username,
password);
        PreparedStatement selectStmt =
conn.prepareStatement(selectIdSQL);

        PreparedStatement updateStmt = 2
conn.prepareStatement(updateFileSQL)) {

        // Prepare and execute the select statement to find the
idfile
        selectStmt.setString(1, selectedFile.getName());
        selectStmt.setString(2, teamId);
        ResultSet rs = selectStmt.executeQuery();

        // Check if the result set has at least one entry
        3
        if (rs.next()) {
            int idFile = rs.getInt("idfile");

            // Prepare and execute the update statement
            updateStmt.setString(1, selectedFile.getName());

```

```

        updateStmt.setInt(2, idFile);
        updateStmt.setString(3, teamId);
        int affectedRows = updateStmt.executeUpdate();

        // Provide feedback about the operation

        if (affectedRows > 0) {

            JOptionPane.showMessageDialog(null, "File replaced
            successfully.", "Success", JOptionPane.INFORMATION_MESSAGE);
        } else {

            JOptionPane.showMessageDialog(null, "No changes were
            made. File not found.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {

        JOptionPane.showMessageDialog(null, "File not found in
        database.", "Error", JOptionPane.ERROR_MESSAGE);
    }
} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, "Database error: " +
    e.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
}
}

```

```

\9\
public void insertFileIntoDatabase(String branchName, byte[]
fileContent) throws SQLException {
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";

    String password = "W45@jqr#8CX";

    String selectBranchIdSQL = "SELECT idbranch, teamId FROM
co_share_branch_projects WHERE branchName = ?";
    String insertFileSQL = "INSERT INTO co_share_file_projects
(fileName, fileType, fileSize, dateUploaded, ownerUsername, branchId,
version, description, teamID, fileContent) VALUES (?, ?, ?, ?, ?, ?,
?, ?, ?, ?)";

    try (Connection conn = DriverManager.getConnection(url, username,
password);
        PreparedStatement selectStmt =

conn.prepareStatement(selectBranchIdSQL);
        PreparedStatement insertStmt =

```



```

conn.prepareStatement(insertFileSQL)) {

    selectStmt.setString(1, branchName);
    ResultSet rs = selectStmt.executeQuery();

    3
    if (rs.next()) {
        int branchId = rs.getInt("idbranch");
        int teamID = rs.getInt("teamId");

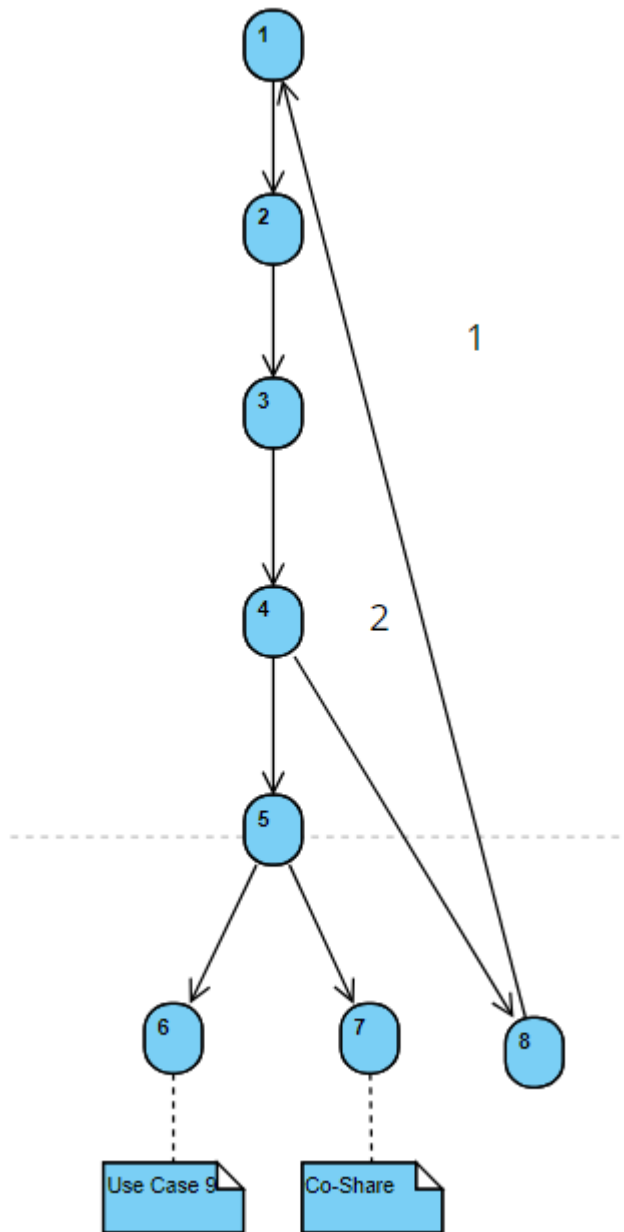
        insertStmt.setString(1, selectedFile.getName());
        insertStmt.setString(2,
getFileExtension(selectedFile.getName()));
        insertStmt.setLong(3, selectedFile.length());
        insertStmt.setTimestamp(4, new Timestamp(new
Date().getTime()));
        insertStmt.setString(5, user.getUsername());
        insertStmt.setInt(6, branchId);
        insertStmt.setString(7, "0.1");

        insertStmt.setNull(8, Types.VARCHAR);
        insertStmt.setString(9, this.teamID);
        insertStmt.setBytes(10, fileContent);

        insertStmt.executeUpdate();
        JOptionPane.showMessageDialog(null, "File successfully
uploaded.", "Success", JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(null, "Branch not found.",
>Error", JOptionPane.ERROR_MESSAGE);
    }
}
}
}

```

ακολουθεί ο γράφος για το ActionListener που γίνεται triggered όταν πατηθεί το πλήκτρο 'MyTeam'.



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 8 - 8 + 2 = 2$  περιοχές, όπως φαίνεται και  
 στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
 συντομότερο έγκυρο μονοπάτι:

**M1: 1-2-3-4-8**

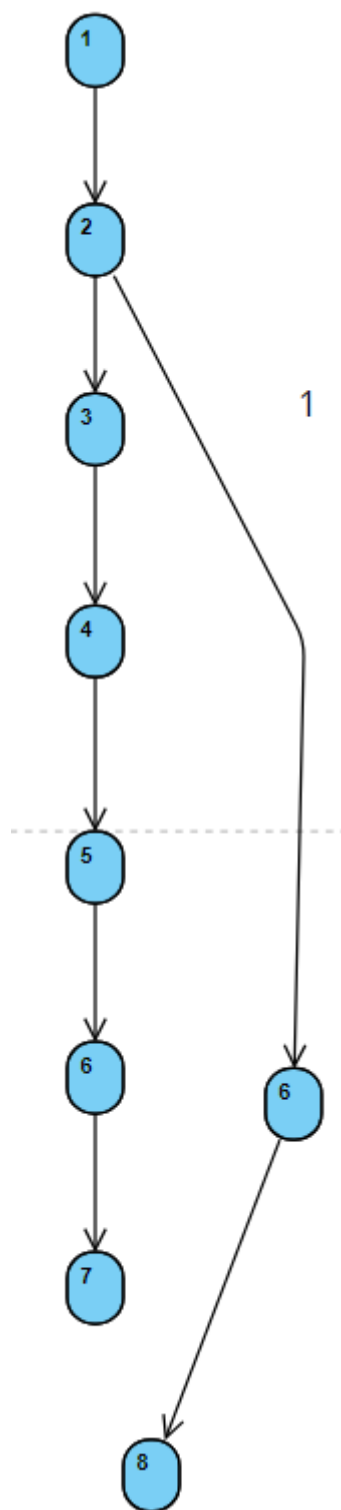
Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
 ακμών στο M1.

M2: 1-2-3-4-5-6 (Alternative Flow 4)

M3: 1-2-3-4-5-7 (steps of Main Flow)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	projectId.trim() = "" or <u>wrong</u>	Message: Invalid Project Id . Please try again
M2	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	userExistsInTeamProjects = <u>true</u>	No: studentFindTeam Panel (Use Case 9)
M3	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	userExistsInTeamProjects = <u>true</u>	Yes: Open StudentTeam Panel

Ακολουθεί ο γράφος για τη Συνάρτηση userExistsInTeamProjects



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 8 - 9 + 2 = 1$  περιοχή, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

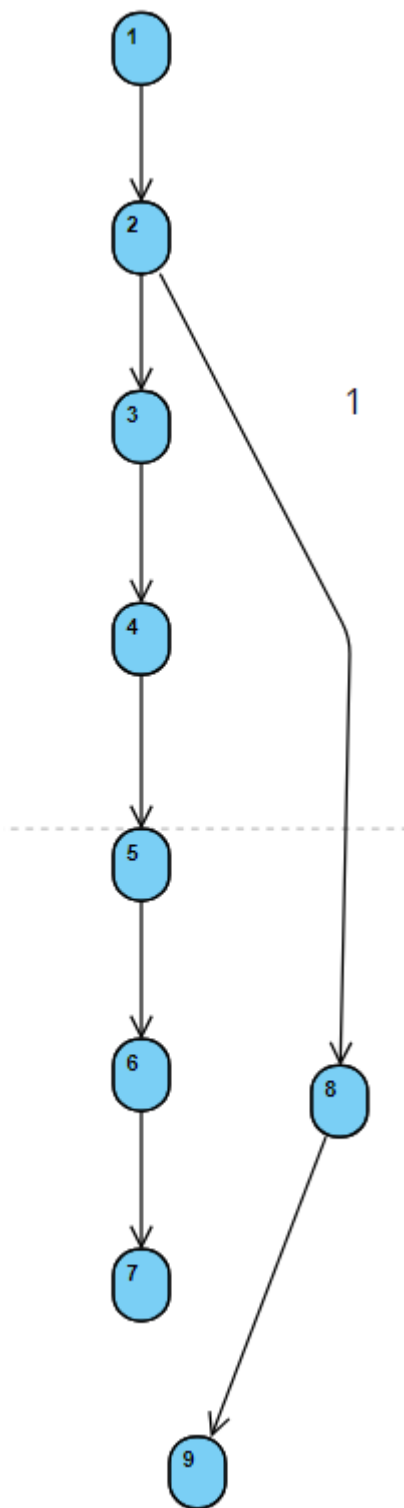
M1: 1-2-6-8

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3-4-5-7 (**steps of Main Flow**)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος σύνδεσης στη Βάση	Connenction con <u>= null</u>	Ex.PrintStackTrace
M2	Έλεγχος αν ο Φοιτητής ανήκει στην Ομάδα με το δοσμένο Id	rs.next() && rs.getInt("count") > 0	Return True;

Ακολουθεί ο γράφος για τη Συνάρτηση isValidProjectId



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 8 - 9 + 2 = 1$  περιοχή, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

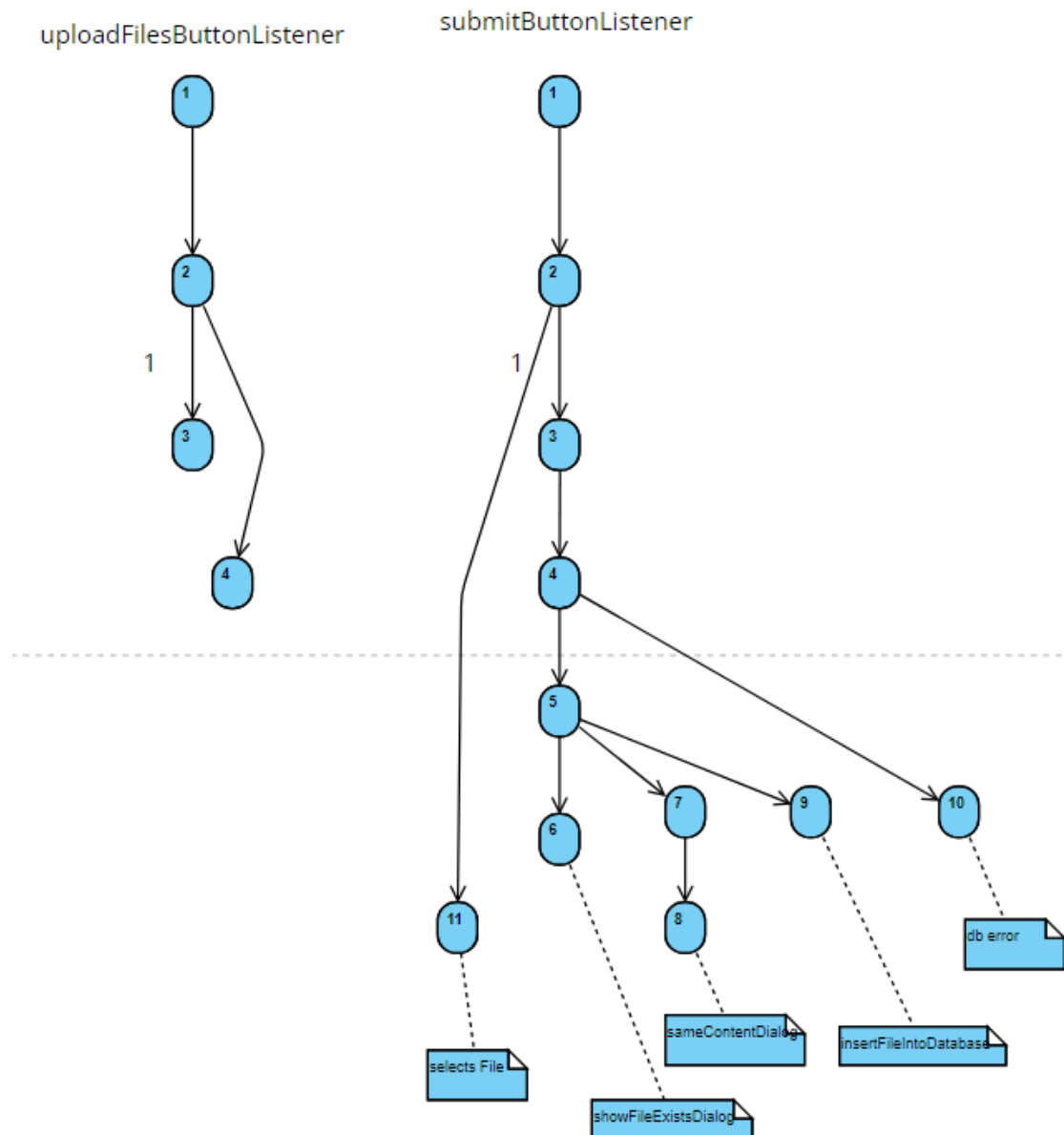
M1: 1-2-8-9

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3-4-5-6-7

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος σύνδεσης στη Βάση	Connenction con $\equiv$ <u>null</u>	Ex.PrintStackTrace
M2	Έλεγχος αν ο Φοιτητής έχει δώσει σωστό Id	rs.next() && rs.getInt("count") > 0	Return True;

Οι γράφοι που ακολουθούν είναι για το uploadFiles και submitButton listeners.



### uploadFileslistener

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 3 - 4 + 2 = 1$  περιοχή, όπως φαίνεται και  
 στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
 συντομότερο έγκυρο μονοπάτι:

M1: 1-2-3



M1: 1-2-4

### **submitButtonlistener**

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 10 - 11 + 2 = 1$  περιοχή, όπως φαίνεται  
και στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
συντομότερο έγκυρο μονοπάτι:

**M1: 1-2-11 (steps of Main Flow)**

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
ακμών στο M1.

**M2: 1-2-3-4-5-6 (Alternative Flow 1)**

**M3: 1-2-3-4-10 (Alternative Flow 2)**

M4: 1-2-3-4-5-7-8

M5: 1-2-3-4-5-9

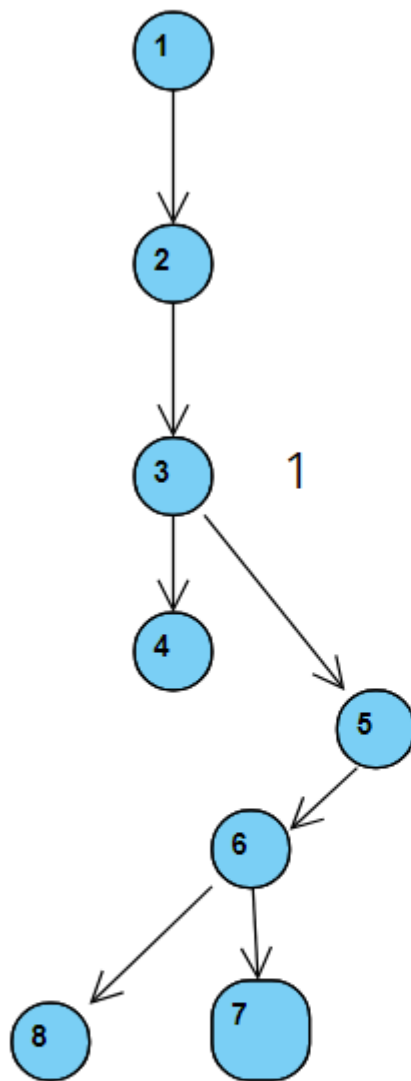
M6: 1-2-3-4-10

Pa th	Περιγ ραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M 1	Έλεγχ ος επιλογ ής φακέλ ου και κλάδο υ	(!fileName.getText().t rim().isEmpty() && !branchName.isE mpty.. <u>Filename or</u> <u>branchName is</u> <u>null</u>	Message: Please select a file and specify Branch.

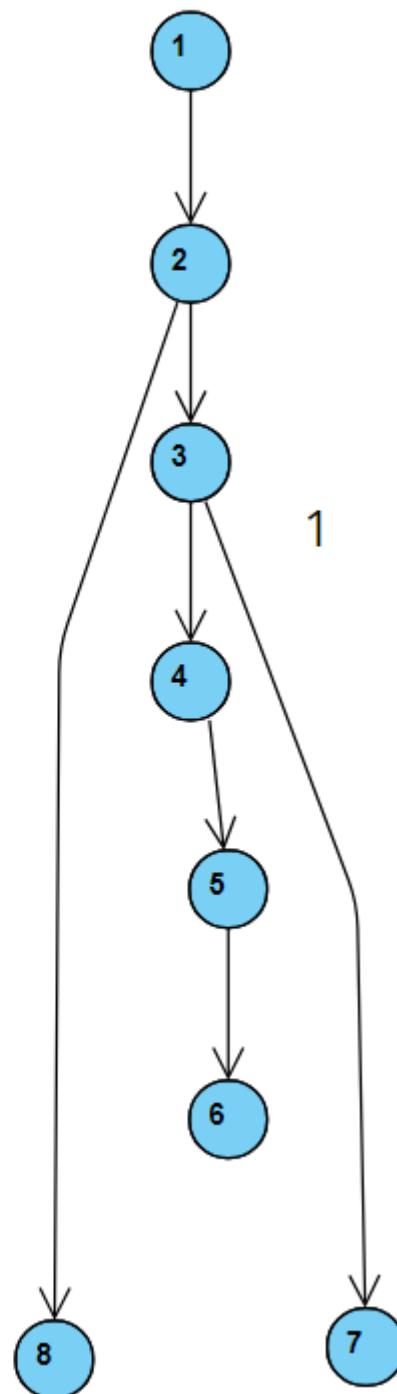
M 2	Έλεγχος της μεθόδου checkFile, σχετικά με το αρχείο προς ανάρτηση	("sameName".equals(checkResult) == <u>true</u> )	dialog.showFileExistsDialog(branchName, fileContent)
M 3	Έλεγχος της μεθόδου checkFile, σχετικά με το αρχείο προς ανάρτηση	("sameContent".equals(checkResult) == <u>true</u> )	dialog.sameContentDialog(branchName, fileContent)

Ο Γράφος για τις μεθόδους checkFileProjects και updateVersion

checkFileProjects



updateVersion



### checkFileProjects

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$

= 7-8+2 = 1 περιοχή, όπως φαίνεται και στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

M1: 1-2-3-4

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3-5-6-7

M3: 1-2-3-5-6-8

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	rsName.next() <u>has some records</u>	Return 'sameName'
M2	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	rsContent.next <u>has some records</u>	Return 'sameContent'

M3	Έλεγχος του προς ανάρτηση αρχείου για ομοιότητα στη Βάση	Else	Return 'allOk'
----	--	------	----------------

## updateVersion

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 7 - 8 + 2 = 1$  περιοχή, όπως φαίνεται και στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

M1: 1-2-8

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

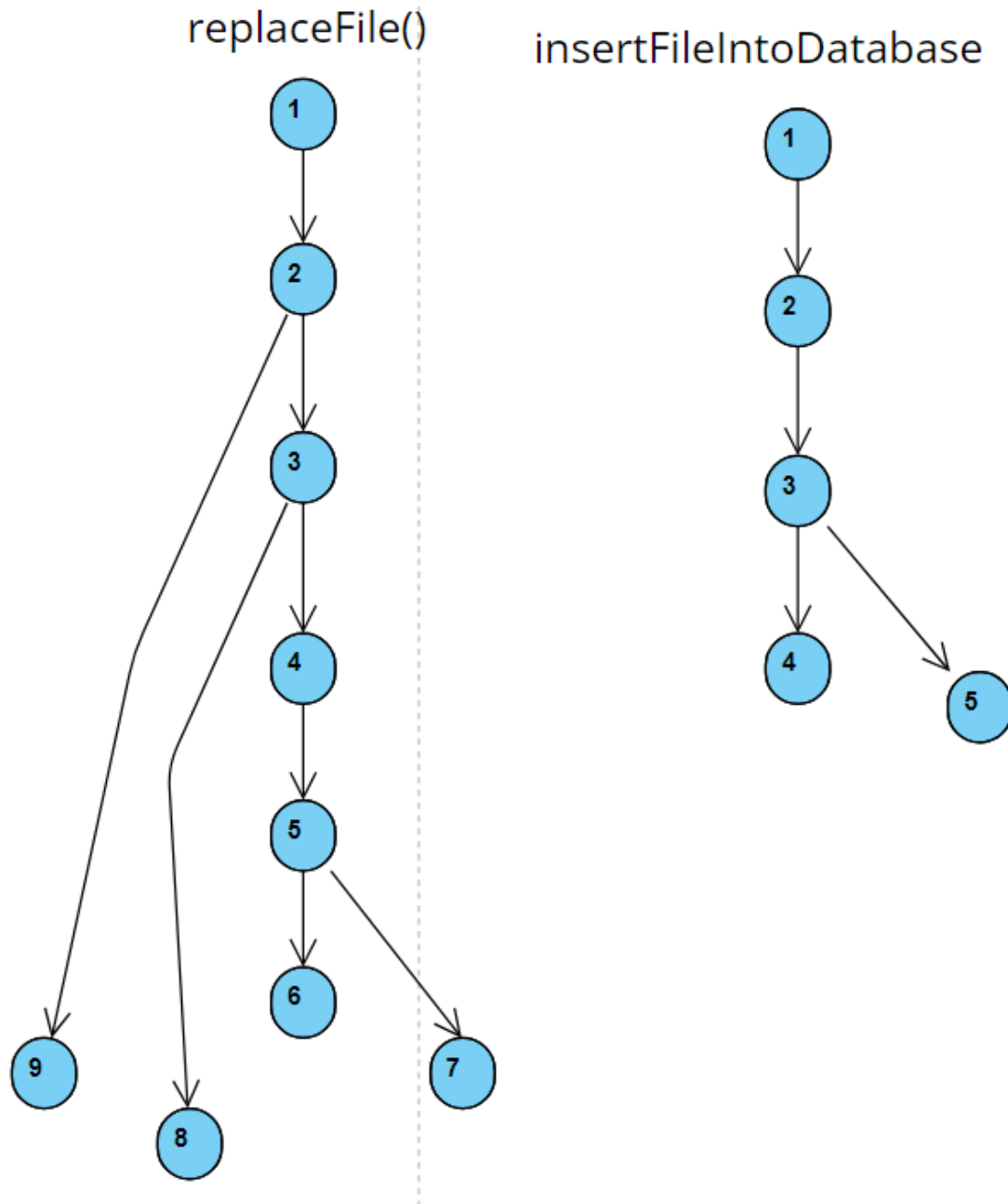
M2: 1-2-3-7

M3: 1-2-3-4-5-6 (**Alternative Flow 1**)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος για σύνδεση στη Βάση	Connection <u>conn</u> = <u>null</u>	Ex.PrintStackTrace

M2	Έλεγχος επιλογής έγκυρου Branch	rsBranch.next() $\equiv$ <u>null</u>	Message: Branch not found
M3	Αναβάθμιση της έκδοσης του αρχείου κατόπιν σωστής επιλογής Branch	rsVersion.next()	Message: New version of the file 'fileName' created successfully.

Ακολουθούν οι γράφοι για τις συναρτήσεις replaceFile και insertFileIntoDatabase.



### replaceFile()

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 8 - 9 + 2 = 1$  περιοχή, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

M1: 1-2-9

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3-8

M3: 1-2-3-4-5-6 (**Alternative Flow 2**)

M4: 1-2-3-4-5-6-7

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα ('Εξοδος Προγράμματος)
M1	Έλεγχος για σύνδεση στη Βάση	Connection conn = <u>null</u>	Ex.PrintStackTrace
M2	Έλεγχος για το αν υπάρχει το αρχείο προς αντικατάσταση στη Βάση.	rs.next() = <u>[]</u>	Message: File not found in Database.



M3	Αντικατάσταση του αρχείου.	<code>affectedRows&gt;0</code> <u><code>= true</code></u>	Message: File replaced successfully.
----	----------------------------	--	--------------------------------------

## Test Case for Use Case 7

```
int response = JOptionPane.showConfirmDialog(this,  
    "This application needs to access your camera and microphone.  
Do you agree?",
```

```
    "Camera and Microphone Access",  
    JOptionPane.YES_NO_OPTION,  
    JOptionPane.QUESTION_MESSAGE);
```

1

2

```
if (response != JOptionPane.YES_OPTION) {  
    JOptionPane.showMessageDialog(this, "Access denied. Returning to  
Subject Profile.");
```

```
    dispose();  
    new userSubjectProfile(null, user, id);
```

3

```
} else {
```

```
    camera.openCamera();
```

\4\

```
}
```

```
cameraLabel.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        super.mouseClicked(e);
```

1

2

```
        if (capture != null && capture.isOpened()) {
```

\3\

```
            camera.releaseCamera();
```

```
        } else {
```

```
            camera.openCamera();
```

\4\

```
        }
```

```
    }
```

```
});
```

```
shareScreenLabel.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        super.mouseClicked(e);  
        new ScreenSharing(null, user, id);  
    }  
});
```

```

public synchronized void openCamera() {
    releaseCamera(); // Release previous camera if opened

    System.setProperty("org.opencv.videoio.registry", "ffmpeg");
    System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

    capture = new VideoCapture(0);

    new Thread(() -> {
        try {
            while (capture != null && capture.isOpened()) {
                Mat frame = new Mat();

                if (capture.read(frame)) {
                    ImageIcon imageIcon = new
ImageIcon(Mat2BufferedImage(frame));

                    speakerLabel.setIcon(imageIcon);
                    speakerLabel.setSize(imageIcon.getIconWidth(),
imageIcon.getIconHeight());
                    Thread.sleep(50); // Control frame rate
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            releaseCamera();
        }
    }).start();

    ImageIcon camOpenLabelIcon = new ImageIcon("src/video-
camera.png");
    cameraLabel.setIcon(camOpenLabelIcon);
}

```

\7\

```

public synchronized void releaseCamera() {
    1
    2
    if (capture != null && capture.isOpened()) {
        capture.release();
        speakerLabel.setIcon(null);
        ImageIcon camCloseIcon = new ImageIcon("src/no-speak.png");
        speakerLabel.setIcon(camCloseIcon);
    }
    3
    ImageIcon camCloseLabelIcon = new ImageIcon("src/no-
video.png");
    cameraLabel.setIcon(camCloseLabelIcon);
}

public synchronized void openMicrophone() {
    releaseMicrophone(); // Release previous microphone if opened
    1

    AudioFormat format = new AudioFormat(16000, 16, 1, true, false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class,
format);

    try {
        line = (TargetDataLine) AudioSystem.getLine(info);
        line.open(format);
        line.start();

        new Thread(() -> {
            2
            try {
                byte[] buffer = new byte[1024];
                while (line.isOpen()) {
                    3
                    int bytesRead = line.read(buffer, 0,
buffer.length);
                }
            } catch (Exception e) {
                4
                e.printStackTrace();
            } finally {
                5
                releaseMicrophone();
            }
        }).start();
    } catch (Exception e) {
        6
        e.printStackTrace();
    }
    ImageIcon micOpenLabelIcon = new ImageIcon("src/microphone-black-

```

```
shape.png");  
    microphoneLabel.setIcon(micOpenLabelIcon);  
}
```

7

\5\

```
public synchronized void releaseMicrophone() {  
    1  
    if (line != null) {  
        line.stop();  
        line.close();  
        System.out.println("Microphone released");  
        ImageIcon micOpenLabelIcon = new ImageIcon("src/microphone-  
off.png");  
        microphoneLabel.setIcon(micOpenLabelIcon);  
    }  
}
```

2

```
public void populateWindows(JComboBox comboBox1) {  
    User32.INSTANCE.EnumWindows(new WinUser.WNDENUMPROC() {  
        @Override
```

1

```

//set the combobox with the window titles found
public boolean callback(WinDef.HWND hWnd, Pointer arg1) {
    2
    if (User32.INSTANCE.IsWindowVisible(hWnd)) {
        char[] windowText = new char[512];
        User32.INSTANCE.GetWindowText(hWnd, windowText, 512);

        String windowTitle =
Native.toString(windowText).trim();

        4
        if (!windowTitle.isEmpty() &&
5
!windowTitle.equals("Program Manager") &&
6
!windowTitle.equals("Εμπειρία εισόδου των Windows") &&
7
!windowTitle.equals("Πυθμίσεις") && !windowTitle.equals("Attending
Lecture for Subject:")) {
            SwingUtilities.invokeLater(() ->
comboBox1.addItem(windowTitle));
        }

        return true; // continue enumeration 9
    }
}, null);
}

```

```

comboBox1.addActionListener(e -> {
    1
    if (updateTimer != null) {
        updateTimer.stop(); 2
    }

    String selectedTitle = (String) comboBox1.getSelectedItem();
    HWND hWnd = ss.findWindowByTitle(selectedTitle); 3

    // Define the action to be performed at each timer tick
    ActionListener updateAction = new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            BufferedImage capturedImage = captureWindow(hWnd);

```

4

```

        if (capturedImage != null) {
            BufferedImage sharpenedImage =
applySharpeningFilter(capturedImage);

            int targetWidth = 900;

            int targetHeight = 650;

            BufferedImage resizedImage =
resizeImage(sharpenedImage, targetWidth, targetHeight);

            // Now set the resized image as the icon of the label
            ImageIcon icon = new ImageIcon(resizedImage);
            screenShareLabel.setIcon(icon);
        }
    };

    // Set the timer to update every 100 milliseconds

    updateTimer = new Timer(100, updateAction);
    updateTimer.start();
});

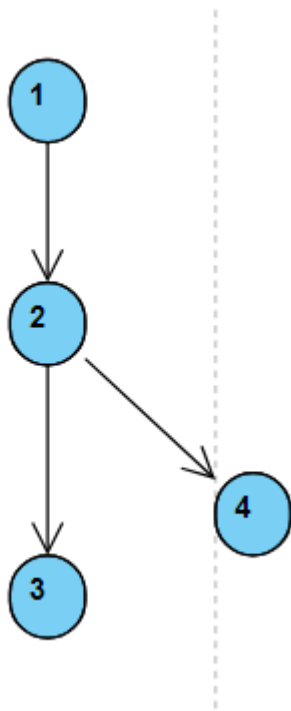
```

5

6

**Για τις μεθόδους `resizeImage`, `applySharpeningFilter`, `captureWindow`, `findWindowByTitle` δεν χρειάστηκε να γίνει κάποια extra αρίθμηση στη ροή του κώδικά τους καθώς δεν περιέχουν κάποιο statement που να οδηγεί σε διαφορετική ροή κώδικα. Ο κώδικάς τους είναι ενιαίος και ακολουθιακός.**

**Ο γράφος για την αποδοχή χρήσης κάμερας και μικροφώνου**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 3 - 4 + 2 = 1$  περιοχή, όπως φαίνεται και  
 στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
 συντομότερο έγκυρο μονοπάτι:

M1α: 1-2-3

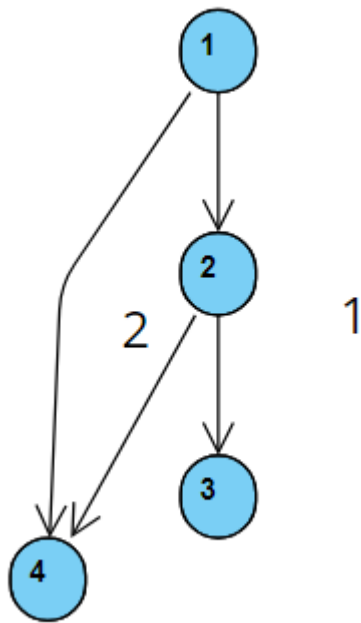
M1β: 1-2-4 (**step for Main Flow**)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1α	Έλεγχος για αποδοχή δικαιωμάτων	(response != JOptionPane.YES_OPTION	New UserSubjectProfile



M2 β	Έλεγχος για αποδοχή δικαιωμάτ ων	Else statement triggered	Camera.openCame ra()
---------	---	-----------------------------	-------------------------

**Ο Γράφος για το cameraLabel listener:**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 4 - 4 + 2 = 2$  περιοχές, όπως φαίνεται και  
 στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
 συντομότερο έγκυρο μονοπάτι:

**M1: 1-4 (steps for Main Flow)**

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
 ακμών στο M1.

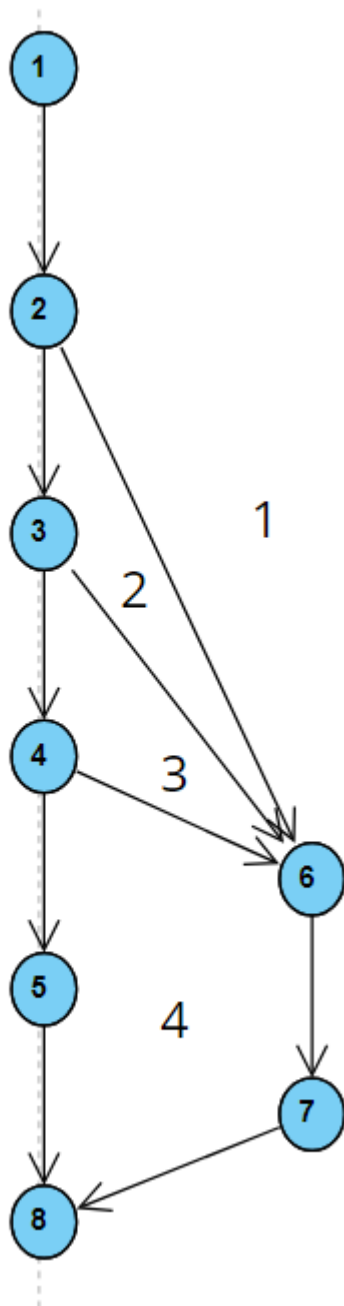
**M2: 1-2-3 (steps for Main Flow)**

**M3: 1-2-4 (steps for Main Flow)**

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος για ύπαρξη γραμμής	<code>capture != null</code>	<code>Camera.openCamera()</code>

	για κάμερα		
M2	Έλεγχος για ύπαρξη γραμμής κάμερας η οποία είναι και ανοιχτή	<code>capture != null &amp;&amp; capture.isOpenen d()</code>	<code>Camera.releaseCame ra()</code>
M3	Έλεγχος για ύπαρξη γραμμής κάμερας η οποία είναι κλειστή	<code>capture != null &amp;&amp; capture.isOpenen d()</code>	<code>Camera.openCamera()</code>

**Ο Γράφος για τη μέθοδο openCamera():**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 10 - 8 + 2 = 4$  περιοχές, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
συντομότερο έγκυρο μονοπάτι:

M1: 1-2-6-7-8

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
ακμών στο M1.

M2: 1-2-3-6-7-8

M3: 1-2-3-4-6-7-8

M4: 1-2-3-4-5-8 (steps for Main Flow)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος για ύπαρξη γραμμής για κάμερα	<code>capture != null</code>	<code>e.printStackTrace()</code> <code>releaseCamera()</code>
M2	Έλεγχος για ύπαρξη γραμμής κάμερας η οποία είναι και ανοιχτή	<code>capture != null &amp;&amp; capture.isOpened()</code>	<code>e.printStackTrace()</code> <code>releaseCamera()</code>
M3	Έλεγχος για αναγνώριση της ανοιχτής γραμμής	<code>capture.read(frame)</code>	<code>e.printStackTrace()</code> <code>releaseCamera()</code>

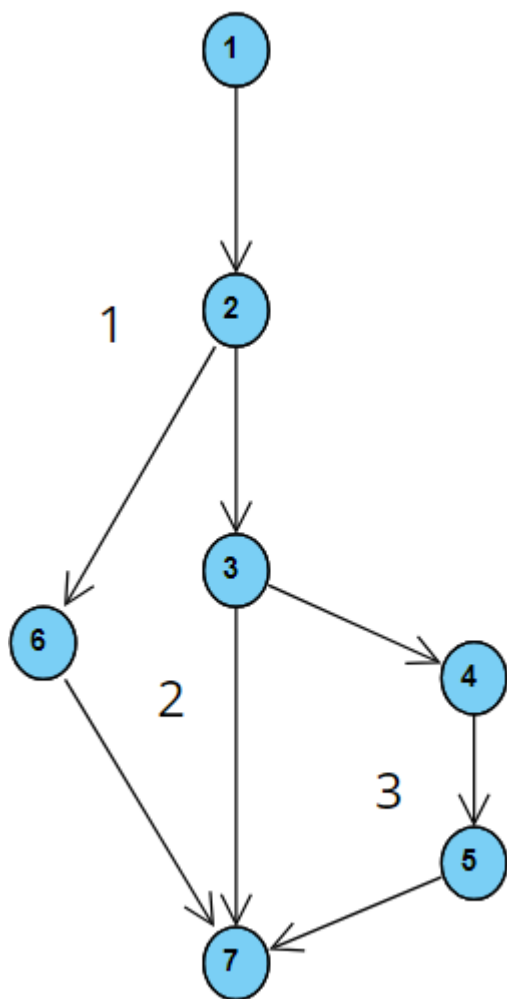
M4	Έλεγχος για ύπαρξη γραμμής κάμερας η οποία είναι και ανοιχτή και μπορεί να αναγνωρισθεί	<code>capture != null &amp;&amp; capture.isOpened() capture.read(frame)</code>	<code>Camera.openCamera() cameraLabel.setIcon()</code>
----	---	--	--

**Ο Γράφος για τη μέθοδο `releaseCamera()`:**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 2 - 3 + 2 = 1$  περιοχή, όπως φαίνεται και  
στον γράφο.

**Ο Γράφος για τη μέθοδο openMicrophone():**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 8 - 7 + 2 = 3$  περιοχές, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
συντομότερο έγκυρο μονοπάτι:

M1α: 1-2-6-7

M1β: 1-2-3-7 (**steps for Main Flow**)

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
ακμών στο M1.

M2: 1-2-3-4-5-7

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα
------	-----------	-------------------	------------------------

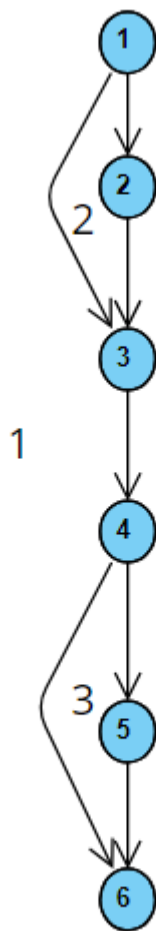


			(Έξοδος Προγράμματος)
M1 α	Δημιουργί α γραμμής για μικρόφων ο	<code>line = (TargetDataLine) AudioSystem.getLine( info); line.open(format); line.start();</code>	<code>e.printStackTrace() microphoneLabel.se tIcon</code>
M1 β	Έλεγχος για ύπαρξη γραμμής μικροφών ου η οποία είναι κλειστή	<code>line.isOpen()</code>	<code>microphoneLabel.se tIcon</code>
M2	Έλεγχος για αναγώρισ η της ανοιχτής γραμμής	<code>line.isOpen()</code>	<code>e.printStackTrace() releaseCamera()</code>

**Ο γράφος για τη συνάρτηση populateWindows():**



**Ο Γράφος για το comboBox1 listener:**



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 7 - 6 + 2 = 3$  περιοχές, όπως φαίνεται και  
στον γράφο.

Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το  
συντομότερο έγκυρο μονοπάτι:

M1: 1-3-4-6

Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη  
ακμών στο M1.

M2: 1-2-3-4-5-6 (**Alternative flow 1**)

M3: 1-2-3-4-6

M4: 1-3-4-5-6

Pat h	Περιγρα φή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
----------	---------------	----------------------	---

M1	Ο μετρητής έχει ξεκινήσει την εύρεση παραθύρων, έχει απαθανατίσει τις οθόνες, ελέγχει για ύπαρξη ανοιχτής γραμμής παραθύρων η οποία είναι κενή	<code>capturedImage != null</code>	Ο μετρητής ξεκινάει εκ νέου από την αρχή
M2	Ο μετρητής έχει ξεκινήσει την εύρεση παραθύρων, έχει απαθανατίσει τις οθόνες, ελέγχει για ύπαρξη ανοιχτής	<code>capturedImage != null</code>	<code>shareLabel.setIcon(resizedImage)</code> Ο μετρητής ξεκινά εκ νέου από την αρχή.

	γραμμής παραθύρ ων η οποία είναι ανοιχτή		
M3	Έλεγχος για αναγώρις η της ανοιχτής γραμμής	<code>capture.read(fr ame)</code>	<code>e.printStackTrace() releaseCamera()</code>