

Test Case for Use Case 3

```
public teacherSendMessage(JFrame parent, User user) {
    this.user = user;
    setTitle("My Messages");
    setContentPane(sendMessagePanel);
    setMinimumSize(new Dimension(1300, 800));
    setLocationRelativeTo(parent);
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    setVisible(true);

    sendButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            dispose(); \1\

            sendMessage(); \2\
            new secretarySendMessage(null, user);
        }
    });

    addWindowListener(new WindowAdapter() {
        @Override
        public void windowClosing(WindowEvent e) {
            new teacherMenuu(null, user);
        }
    });

    showInMessages(); \3\

    showOutMessages(); \4\
}
```

\1\

```
public void dispose() {  
    validateVM(); 1  
    shutdown = true;  
    try {  
        JDWP.VirtualMachine.Dispose.process(vm);  
  
    } catch (JDWPException exc) { 2  
        throw exc.toJDIException();  
    }  
  
    target.stopListening(); 3  
}
```

Για τη συνάρτηση `validateVM()` δεν χρειάστηκε κάποια απαρίθμηση στη ροή του κώδικα καθώς δεν υπάρχει κάποιο `statement` που να αλλάζει τη ροή του κώδικα.

\2\

```
private void sendMessage() {  
    String recipient = recipientField.getText().trim();  
    String messageText = messageBody.getText();  
    String title = titleField.getText();  
  
    Date currentDate = Date.valueOf(LocalDate.now()); 1  
  
    2 3  
    if (recipient.isEmpty() && (!toAll.isSelected()) &&  
    4 5  
    (!toAllTeachers.isSelected()) && (!toAllStudents.isSelected())) {  
        JOptionPane.showMessageDialog(this, "Please select a  
        recipient.");  
        return; 6  
    }  
}
```

7

```
if (title.isEmpty()) {  
    JOptionPane.showMessageDialog(this, "Please give a title to  
your message.");
```

```
    return; 8
```

```
}
```

9

```
if (messageText.isEmpty()) {  
    JOptionPane.showMessageDialog(this, "Your message doesn't have  
any body.");
```

```
    return; 10
```

```
}
```

```
try {  
    // Establish connection to your database  
    Connection connection =  
    DriverManager.getConnection("jdbc:mysql://localhost:3306/softengin23_  
24", "root", "W45@jqr#8CX"); 11
```

12

```
if(toAll.isSelected()){  
    String getUsersQuery = "SELECT username FROM users WHERE  
username NOT LIKE ?";  
    PreparedStatement getUsersStatement =  
    connection.prepareStatement(getUsersQuery);  
    getUsersStatement.setString(1, user.username);  
  
    ResultSet resultSet = getUsersStatement.executeQuery();
```

```
    // Insert message for each user 13
```

14

```
while (resultSet.next()) {  
    String recipientUsername =  
    resultSet.getString("username");  
  
    // Prepare the SQL statement  
    PreparedStatement preparedStatement =  
    connection.prepareStatement(  
        "INSERT INTO my_messages (sender, recipient,  
message_text, title, date) VALUES (?, ?, ?, ?, ?)");  
  
    // Set parameters  
    preparedStatement.setString(1, user.getUsername());  
    preparedStatement.setString(2, recipientUsername);  
    preparedStatement.setString(3, messageText);  
    preparedStatement.setString(4, title);  
    preparedStatement.setDate(5, currentDate);
```

```

        // Execute the statement
        preparedStatement.executeUpdate();

        // Close the prepared statement
        preparedStatement.close(); 15
    }

    // Close result set and statement
    resultSet.close();

    getUsersStatement.close(); 16
}

17
else if(toAllTeachers.isSelected()){
    String getUsersQuery = "SELECT type,username FROM users
WHERE type NOT LIKE 'student'";
    PreparedStatement getUsersStatement =
connection.prepareStatement(getUsersQuery);

    ResultSet resultSet = getUsersStatement.executeQuery();

    // Insert message for each user 18

19
    while (resultSet.next()) {
        String recipientUsername =
resultSet.getString("username");

        // Prepare the SQL statement
        PreparedStatement preparedStatement =
connection.prepareStatement(
            "INSERT INTO my_messages (sender, recipient,
message_text, title, date) VALUES (?, ?, ?, ?, ?)");

        // Set parameters
        preparedStatement.setString(1, user.getUsername());
        preparedStatement.setString(2, recipientUsername);
        preparedStatement.setString(3, messageText);
        preparedStatement.setString(4, title);
        preparedStatement.setDate(5, currentDate);

        // Execute the statement
        preparedStatement.executeUpdate();

        // Close the prepared statement
        preparedStatement.close(); 20
    }

    // Close result set and statement
    resultSet.close();

```

```
        getUsersStatement.close(); 21
    }
```

22

```
        else if(toAllStudents.isSelected()){
            String getUsersQuery = "SELECT type,username FROM users
WHERE type NOT LIKE 'teacher'";
            PreparedStatement getUsersStatement =
connection.prepareStatement(getUsersQuery);

            ResultSet resultSet = getUsersStatement.executeQuery();
```

```
            // Insert message for each user 23
```

24

```
            while (resultSet.next()) {
                String recipientUsername =
resultSet.getString("username");

                // Prepare the SQL statement
                PreparedStatement preparedStatement =
connection.prepareStatement(
                    "INSERT INTO my_messages (sender, recipient,
message_text, title, date) VALUES (?, ?, ?, ?, ?)");

                // Set parameters
                preparedStatement.setString(1, user.getUsername());
                preparedStatement.setString(2, recipientUsername);
                preparedStatement.setString(3, messageText);
                preparedStatement.setString(4, title);
                preparedStatement.setDate(5, currentDate);

                // Execute the statement
                preparedStatement.executeUpdate();

                // Close the prepared statement

                preparedStatement.close(); 25
            }
```

```
            // Close result set and statement
            resultSet.close();
```

```
            getUsersStatement.close(); 26
```

```
        }
    } else{
```

```
        // Prepare the SQL statement
        PreparedStatement preparedStatement =
connection.prepareStatement(
            "INSERT INTO my_messages (sender, recipient,
message_text, title, date) VALUES (?, ?, ?, ?, ?)");

        // Set parameters
        preparedStatement.setString(1, user.getUsername());
```

```

        preparedStatement.setString(2, recipient);
        preparedStatement.setString(3, messageText);
        preparedStatement.setString(4, title);
        preparedStatement.setDate(5, currentDate);

        // Execute the statement
        preparedStatement.executeUpdate();

        // Close the resources

        preparedStatement.close(); 27
    }
    connection.close();

    // Display a success message
    JOptionPane.showMessageDialog(this, "Message sent
successfully!"); 28
    } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error sending message: "
+ ex.getMessage());
    } 29
} 30

```

\3\

```

private void showInMessages() {
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";
    String password = "W45@jqr#8CX";

    DefaultListModel<String> inListModel = new DefaultListModel<>();

    StringBuilder inListText = new StringBuilder(); 1

    2

    try (Connection connection = DriverManager.getConnection(url,
username, password)) {
        String recipient = user.username;

        String sql = "SELECT sender, recipient, message_text, title,
date " +
            "FROM my_messages " +
            "WHERE recipient = ?";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setString(1, recipient);
    }
}

```

```

ResultSet resultSet = statement.executeQuery(); 3

4
while (resultSet.next()) {
    String sender = resultSet.getString("sender");
    String messageRecipient = resultSet.getString("recipient");
    String messageText = resultSet.getString("message_text");
    String title = resultSet.getString("title");
    String date = resultSet.getString("date");

    inListText.append("<html>");
    inListText.append(sender + " to " + messageRecipient +
"<br>About: " + title + "<br>" );

    String textSub = null; 5

6
    if (messageText.length() > 65) {
        textSub = messageText.substring(0, 65) + "...";

        inListText.append(textSub).append("<br>"); 7
    }
    else{

        inListText.append(messageText).append("<br>"); 8
    }

    inListText.append("<br>On: " + date);
    inListText.append("<br><hr><br></html>");

    inListModel.addElement(inListText.toString());

    inListText.setLength(0); 9

}

resultSet.close();
statement.close();
connection.close();

inList.setModel(inListModel); 10

} catch (SQLException e) {
    e.printStackTrace(); 11
}
} 12

```

\4\

```
private void showOutMessages() {
    String url = "jdbc:mysql://localhost:3306/softengin23_24";
    String username = "root";
    String password = "W45@jqr#8CX";

    DefaultListModel<String> outListModel = new DefaultListModel<>();

    StringBuilder outListText = new StringBuilder(); 1

    2
    try (Connection connection = DriverManager.getConnection(url,
        username, password)) {
        String recipient = user.username;

        String sql = "SELECT sender, recipient, message_text, title,
date " +
            "FROM my_messages " +
            "WHERE sender = ?";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setString(1, recipient);

        ResultSet resultSet = statement.executeQuery(); 3

        4
        while (resultSet.next()) {
            String sender = resultSet.getString("sender");
            String messageRecipient = resultSet.getString("recipient");
            String messageText = resultSet.getString("message_text");
            String title = resultSet.getString("title");
            String date = resultSet.getString("date");

            outListText.append("<html>");
            outListText.append(sender + " to " + messageRecipient +
"<br>About: " + title + "<br>" );

            String textSub = null; 5

            6
            if (messageText.length() > 65) {
                textSub = messageText.substring(0, 65) + "...";

                outListText.append(textSub).append("<br>"); 7
            }
            else{

                outListText.append(messageText).append("<br>"); 8
            }

            outListText.append("<br>On: " + date);
        }
    }
}
```



```

        outListText.append("<br><hr><br></html>");

        outListModel.addElement(outListText.toString());

        outListText.setLength(0); 9

    }

    resultSet.close();
    statement.close();
    connection.close();

    outList.setModel(outListModel); 10

} catch (SQLException e) {
    e.printStackTrace();

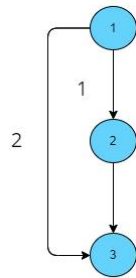
} 11

} 12

```

Για τη συνάρτηση `teacherSendMessage` δεν χρειάστηκε γράφος γιατί εκτελούντε όλες η εντολές και καλούντε όλες οι συναρτήσεις με την σειρά και δεν υπάρχει κάποια συνθήκη.

Παρακάτω παρατίθεται ο Γράφος για την συνάρτηση `dispose()`.



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

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

M1: 1-3 (steps of Main flow)

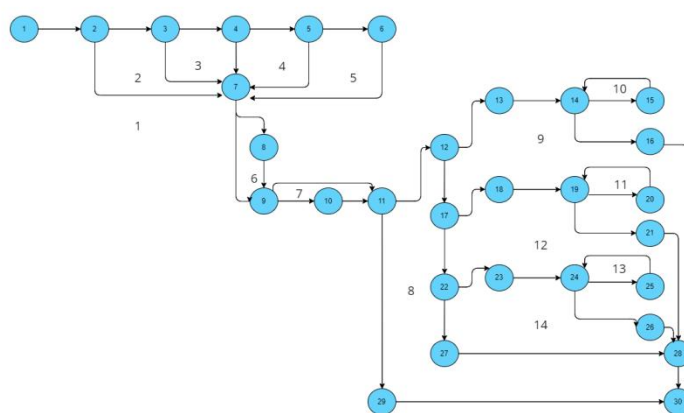
Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα ('Εξοδος Προγράμματος)
M2	Σφάλμα Κατά τον Τερματισμό Virtual Machine	<pre>try { JDWP.VirtualMachine.Dispose.process(vm); }</pre>	<pre>catch (JDWPException exc) { throw exc.toJDIException(); }</pre>

			target.stopListening();
M1	Επιτυχής Τερματισμός Virtual Machine	try { JDWP.VirtualMachine.Dispose.process(vm); }	target.stopListening();

Παρακάτω παρατίθεται ο Γράφος για την συνάρτηση `sendMessage()`.



Σύμφωνα με τον τύπο: $V(g) = e - n + 2 = 42 - 30 + 2 = 14$ περιοχές, όπως φαίνεται και στον γράφο.

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

M1: 1-2-7-9-11-29-30

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

M2: 1-2-7-9-11-12-17-22-27-28-30 (**steps of Main Flow**)

M3: 1-2-7-8-9-11-12-17-18-19-20-19-21-28-30 (**Alternative Flow 1**)

ή

M4: 1-2-7-9-10-11-12-17-18-19-20-19-21-28-30 (**Alternative Flow 1**)

M5: 1-2-3-4-5-6-7-9-11-12-17-22-27-28-30 (**Alternative Flow 2**)

M6: 1-2-7-9-11-12-17-22-23-24-25-24-26-28-30 **σχεδιαστική**

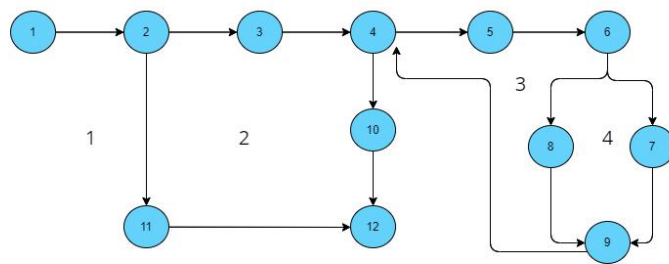
παράλειψη έστω

ότι υπάρχει ακμή ανάμεσα από το 12 και το 17 που να ελέγχει για το πόσα αρχεία έχουν επιλεχθεί (Alternative Flow 3)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M2	Ο Καθηγητής έχει κάνει όλα τα βήματα σωστά και το μήνυμα εστάλη επιτυχώς	Η περίπτωση ελέγχου είναι ο καθηγητής να έχει ακολουθήσει όλα τα βήματα σωστά	"Message sent successfully!"
M4	Ο Καθηγητής επιλέγει τον παραλήπτη που αυτή τη φορά είναι η ολομέλεια των Καθηγητών αλλά ξεχνά να συντάξει το κείμενο	<pre> if (messageText.isEmpty()) { JOptionPane.showMessageDialog(this, "Your message doesn't have any body."); return; } </pre>	"Your message doesn't have any body."
M3	Ο Καθηγητής επιλέγει τον παραλήπτη που αυτή τη φορά είναι η ολομέλεια των Καθηγητών αλλά ξεχνά να συντάξει τον	<pre> if (title.isEmpty()) { JOptionPane.showMessageDialog(this, "Please give a title to your message."); return; } </pre>	"Please give a title to your message."

	Τίτλο		
M5	Ο Καθηγητής αμελεί να επιλέξει τον παραλήπτη	<pre>(!toAllTeachers.isSelected()) && (!toAllStudents.isSelected())) { JOptionPane.showMessageDialog og(this, "Please select a recipient."); return; }</pre>	‘You must select a recipient please..’
M6	Ο Καθηγητής επιλέγει τον παραλήπτη που αυτή τη φορά είναι η ολομέλεια των Φοιτητών αλλά επισυνάπτε ι 3 αρχεία για απόστολή	<p>σχεδιαστική παράλειψη έστω ότι υπάρχει ακμή ανάμεσα από το 12 και το 17 που να ελέγχει για το πόσα αρχεία έχουν επιλεχθεί</p>	‘Your Message cant contain more than 2 files!’.

Παρακάτω παρατίθεται ο Γράφος για την συνάρτηση `showInMessages()`.



Σύμφωνα με τον τύπο: $V(g) = e - n + 2 = 14 - 12 + 2 = 4$ περιοχές, όπως φαίνεται και στον γράφο.
Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

M1: 1-2-11-12

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

M2: 1-2-3-4-10-12

M3: 1-2-3-4-5-6-7-9-4-10-12 (**steps of Main flow**)

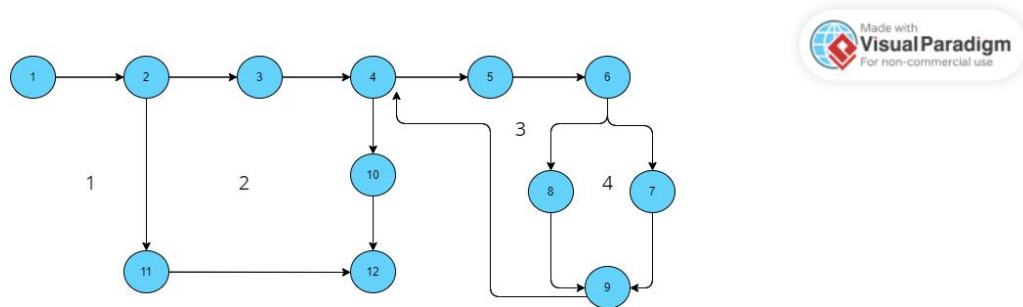
ή

M4: 1-2-3-4-5-6-8-9-4-10-12 (**steps of Main flow**)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M3	Η συνάρτηση εκτελείται χωρίς προβλήματα	<pre> if (messageText.length() > 65) { textSub = messageText.substring(0, 65) + "..."; </pre>	Τα μηνύματα εμφανίζονται στη γραφική διεπαφή

	τα και τα μηνύματα εμφανίζο νται στη γραφική διεπαφή	<code>inListText.append(textSub).app end("
"); }</code>	
M4	Η συνάρτησ η εκτελείτ αι χωρίς προβλήμα τα και τα μηνύματα εμφανίζο νται στη γραφική διεπαφή	<code>else{ inListText.append(messageText) .append("
"); }</code>	τα μηνύματα εμφανίζοντα ι στη γραφική διεπαφή

Παρακάτω παρατίθεται ο Γράφος για την συνάρτηση `showOutMessages()`.



Σύμφωνα με τον τύπο: $V(g) = e - n + 2 = 14 - 12 + 2 = 4$ περιοχές, όπως φαίνεται και στον γράφο.
Εφαρμόζουμε τον αλγόριθμο για εύρεση μονοπατιών. Ξεκινάμε με το συντομότερο έγκυρο μονοπάτι:

M1: 1-2-11-12

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

M2: 1-2-3-4-10-12

M3: 1-2-3-4-5-6-7-9-4-10-12 (steps of Main flow)

ή

M4: 1-2-3-4-5-6-8-9-4-10-12 (steps of Main flow)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M3	Η συνάρτηση	if (messageText.length() > 65) { textSub	= τα μηνύματα εμφανίζονται στη

	εκτελείται χωρίς προβλήματα και τα μηνύματα εμφανίζονται στη γραφική διεπαφή	<pre> messageText.substring(0, 65) + "..."; outListText.append(textSub).append("
"); } </pre>	γραφική διεπαφή
M4	Η συνάρτηση εκτελείται χωρίς προβλήματα και τα μηνύματα εμφανίζονται στη γραφική διεπαφή	<pre> else{ outListText.append(messageText).append("
"); } </pre>	τα μηνύματα εμφανίζονται στη γραφική διεπαφή