

## Test Case for Use Case 10

```
public void drawCalendar(JTable calendarTable, User user) {
    LocalDate DcurrentDate = LocalDate.now();

    Projects project = new Projects();
    project.loadProjectDates(user, projectDetails);

    Announcements ann = new Announcements();
    ann.loadAnnouncementDates(announcementDetails);

    Meetings meet = new Meetings();
    meet.loadUserSubjectMeetings(user, detailsMeetings);

    loadUserReminders(user);

    // retrieve year and month
    YearMonth currentYearMonth = YearMonth.from(DcurrentDate);
    int year = currentYearMonth.getYear();
    int month = currentYearMonth.getMonthValue();

    // set the table and the headers
    String[] cols = {"Sun", "Mond", "Tues", "Wedn", "Thur", "Frid",
    "Sat"};
    int numRows = 6; // Max num of rows in a month
    DefaultTableModel modelCal = new DefaultTableModel(cols, numRows)
    {
        @Override
        public boolean isCellEditable(int row, int column) {
            // Make cell non-editable
            return false;
        }
    };

    calendarTable.setModel(modelCal);

    // first day of the month
    LocalDate firstDayOfMonth = LocalDate.of(year, month, 1);

    // get the day of the week for the first day of the month
    int startDayOfWeek = firstDayOfMonth.getDayOfWeek().getValue() %
    7;

    // number of days in the month
    int daysInMonth = currentYearMonth.lengthOfMonth();

    // populate the table
    int row = 0;
    int col = startDayOfWeek;

    for (int day = 1; day <= daysInMonth; day++) {
        modelCal.setValueAt(day, row, col);

        if (col == 6) {
            // move to the next row if it's Saturday
            row++;
        }
    }
}
```

```

        col = 0;
    } else {
        // move to the next column
        col++;
    }
}

calendarTable.setDefaultRenderer(Object.class, new
DefaultTableCellRenderer() {
    @Override
    public Component getTableCellRendererComponent(JTable table,
Object value,
boolean
isSelected, boolean hasFocus,
int row, int
column) {
        // invoke super method to get the default renderer
        JLabel c = (JLabel)
super.getTableCellRendererComponent(table, value, isSelected,
hasFocus, row, column);
        LocalDate cellDate = firstDayOfMonth.plusDays((row * 7) +
column - startDayOfWeek);
        String dayOfWeekFull =
cellDate.getDayOfWeek().getDisplayName(TextStyle.FULL,
Locale.ENGLISH);

        boolean hasProject =
projectDetails.containsKey(cellDate);
        boolean hasAnnouncement =
announcementDetails.containsKey(cellDate);
        boolean hasMeeting =
detailsMeetings.containsKey(dayOfWeekFull);

        // retrieve the value in the cell
        Object cellValue = table.getValueAt(row, column);

        // set the color cell of today's day to gray
        if (cellValue != null && cellValue instanceof Integer &&
((int) cellValue) == LocalDate.now().getDayOfMonth()) {
            c.setBackground(new Color(205, 204, 204));
        } else {
            // reset background color for the rest of the cells
            c.setBackground(Color.WHITE);
        }
    }
}

```

```

12         if (hasProject && hasAnnouncement && hasMeeting) {
13             c.setIcon(new MultipleDotsIcon(Color.BLUE, Color.RED,
14                 Color.GREEN));
15             c.setToolTipText("Multiple events, click for
16                 details");
17         } else if (hasProject) {
18             c.setIcon(new DotIcon(Color.BLUE));
19             c.setToolTipText("Project deadline, click for
20                 details");
21         } else if (hasAnnouncement) {
22             c.setIcon(new DotIcon(Color.RED));
23             c.setToolTipText("Announcement today, click for
24                 details");
25         } else if (hasMeeting) {
26             c.setIcon(new DotIcon(Color.GREEN));
27             c.setToolTipText("Meeting today, click for details");
28         } else {
29             c.setIcon(null);
30         }
31
32         if (userReminders.containsKey(cellDate)) {
33             c.setIcon(new DotIcon(Color.BLACK));
34             String reminderTooltip = "<html>";
35             for (String[] details : userReminders.get(cellDate))
36             {
37                 reminderTooltip += "Reminder: " + details[0] + "
38 - " + details[1] + "<br>";
39             }
40
41             reminderTooltip += "</html>";
42             c.setToolTipText(reminderTooltip);
43         }
44
45         c.setHorizontalAlignment(JLabel.CENTER); // Ensure text
46         is centered
47
48         return c;
49     }

```

```

});

calendarTable.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {

        1
        if (e.getClickCount() == 2) { // Check for double click
            int row = calendarTable.rowAtPoint(e.getPoint());
            int column =
calendarTable.columnAtPoint(e.getPoint());
            LocalDate firstDayOfMonth =
LocalDate.of(currentYearMonth.getYear(),

currentYearMonth.getMonthValue(), 1);
                2
            int startDayOfWeek =
firstDayOfMonth.getDayOfWeek().getValue() % 7;
            LocalDate cellDate = firstDayOfMonth.plusDays((row *
7) + column - startDayOfWeek);
            Reminder reminder = new Reminder();
            reminder.handleReminderInput(cellDate, calendarTable,
user);
        }
    }
});

calendarTable.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {

        1
        if (e.getClickCount() == 1) {
            int row = calendarTable.rowAtPoint(e.getPoint());
            int column =
calendarTable.columnAtPoint(e.getPoint());
            LocalDate cellDate = firstDayOfMonth.plusDays((row *
7) + column - startDayOfWeek);

            String dayOfWeekFull =
                2
cellDate.getDayOfWeek().getDisplayName(TextStyle.FULL,
Locale.ENGLISH);

            3
            if (projectDetails.containsKey(cellDate)) {
                String[] details = projectDetails.get(cellDate);
                String message = "Today is the deadline for the "

+ details[1] + " in " + details[0];
                4
                JOptionPane.showMessageDialog(calendarTable,
message, "Project Deadline", JOptionPane.INFORMATION_MESSAGE);
            }

            5
            if (announcementDetails.containsKey(cellDate)) {
                String[] ann_details =

```

```

announcementDetails.get(cellDate);
    String message = ann_details[1] + " says: " +
6
ann_details[0];
    JOptionPane.showMessageDialog(calendarTable,
message, ann_details[2], JOptionPane.INFORMATION_MESSAGE);
}

7
    if (userReminders.containsKey(cellDate)) {
        List<String[]> rem_details =
8
userReminders.get(cellDate);
        StringBuilder message = new StringBuilder();
9
        for (String[] details : rem_details) {
10
            if (message.length() > 0) {
11
                message.append("\n");
            }
            message.append("Reminder:
12
").append(details[0]).append(" - ").append(details[1]);
        }
        JOptionPane.showMessageDialog(calendarTable,
message.toString(), "Reminders", JOptionPane.INFORMATION_MESSAGE);
    }

13
    if (detailsMeetings.containsKey(dayOfWeekFull)) {
14
        List<String[]> meetings =
detailsMeetings.get(dayOfWeekFull);
        for (String[] meetingDetails : meetings) {
            String status = meetingDetails[4];
            String meetingSubject = meetingDetails[0];
            String startTime = meetingDetails[2];
            String endTime = meetingDetails[3];

            String alertMessage;
15
            if ("canceled".equalsIgnoreCase(status)) {
                alertMessage = "The meeting for " +
16
meetingSubject + " has been canceled.";
            } else if
17

```

```

("postponed".equalsIgnoreCase(status)) {
    alertMessage = "The meeting for " +
meetingSubject + " has been postponed. You will be notified of the
specific day and hour.";
} else {
    alertMessage = "Your meeting for " +
meetingSubject + " is " + status + " for today at " + startTime + " -
" + endTime;
}
OptionPane.showMessageDialog(calendarTable,
alertMessage, "Meeting Status", JOptionPane.INFORMATION_MESSAGE);
});
}

```

\1\

```

public void loadUserSubjectMeetings(User user, Map<String,
List<String[]>> meetingDetails) {
    // SQL query to retrieve subjects for the user
    String subjectSql = "SELECT subject_name FROM my_subjects WHERE
student = ?";
    // SQL query to retrieve meeting details based on subject names
    String meetingSql = "SELECT meetingsAbout, host, every1,
startTime, endTime, status FROM meetings_schedule_subjects WHERE
meetingsAbout = ?";

    Set<String> userSubjects = new HashSet<>();

    // First, collect all subjects related to the user
    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/softengin23_
24", "root", "W45@jqr#8CX");

        PreparedStatement pstmtSubject =
conn.prepareStatement(subjectSql)) {
        pstmtSubject.setString(1, user.getUsername());
        ResultSet rsSubject = pstmtSubject.executeQuery();

        while (rsSubject.next()) {

            userSubjects.add(rsSubject.getString("subject_name"));

```

```

    }
    rsSubject.close();

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

// Then, fetch meeting details for each subject
try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/softengin23_
24", "root", "W45@jqr#8CX");
    PreparedStatement pstmtMeeting =
conn.prepareStatement(meetingSql)) {
    for (String subject : userSubjects) {

        pstmtMeeting.setString(1, subject);
        ResultSet rsMeeting = pstmtMeeting.executeQuery();

        while (rsMeeting.next()) {
            String dayOfWeek = rsMeeting.getString("every1");
            String meetingsAbout =
rsMeeting.getString("meetingsAbout");

            String host = rsMeeting.getString("host");
            String startTime = rsMeeting.getString("startTime");
            String endTime = rsMeeting.getString("endTime");
            String status = rsMeeting.getString("status");

            if (!meetingDetails.containsKey(dayOfWeek)) {

                meetingDetails.put(dayOfWeek, new ArrayList<>());
            }
            meetingDetails.get(dayOfWeek).add(new String[]
{meetingsAbout, host, startTime, endTime, status});

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

private void loadUserReminders(User user) {
    String sql = "SELECT date, remindersTitle, remindersText FROM
reminders WHERE username = ?";
    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/softengin23_

```

```

24", "root", "W45@jqr#8CX");
    PreparedStatement pstmt = conn.prepareStatement(sql)) {
    pstmt.setString(1, user.getUsername());
    ResultSet rs = pstmt.executeQuery();

    while (rs.next()) {
        LocalDate reminderDate =
rs.getDate("date").toLocalDate();
        String title = rs.getString("remindersTitle");
        String text = rs.getString("remindersText");

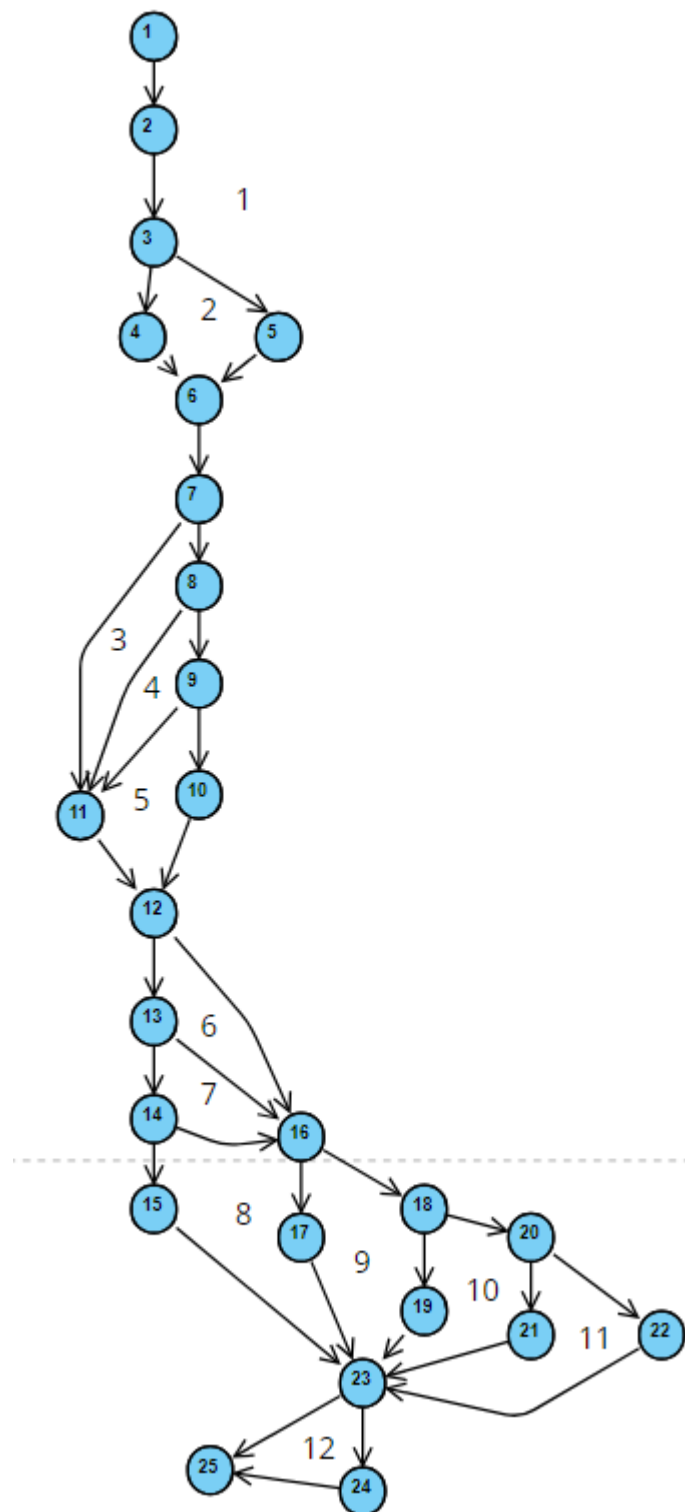
        if (!userReminders.containsKey(reminderDate)) {
            userReminders.put(reminderDate, new ArrayList<>());
        }
        userReminders.get(reminderDate).add(new String[] {title,
text});
    }
} catch (SQLException e) {
    e.printStackTrace();
}

```

**Για τις συναρτήσεις loadAnnouncementDates, loadProjectDates δεν χρειάστηκε να γίνει κάποιος γράφος, αφού η ροή του κώδικά τους είναι ενιαία και συνεχής.**

**Ο Γράφος για τη μέθοδο drawCalendar()**





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

$$= 35-25+2 = 12 \text{ περιοχές,}$$

όπως φαίνεται και στον γράφο.

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

M1: 1-2-3-4-6-7-11-12-16-17-23-24

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

M2: 1-2-3-4-6-7-11-12-16-18-20-21-23-25 (**steps of Main Flow**)

M3: 1-2-3-4-6-7-11-12-16-18-19-23-25 (**Alternative Flow 1**)

M4: 1-3-4-5-6-7-11-12-16-17-23-25 (**Alternative Flow 2**)

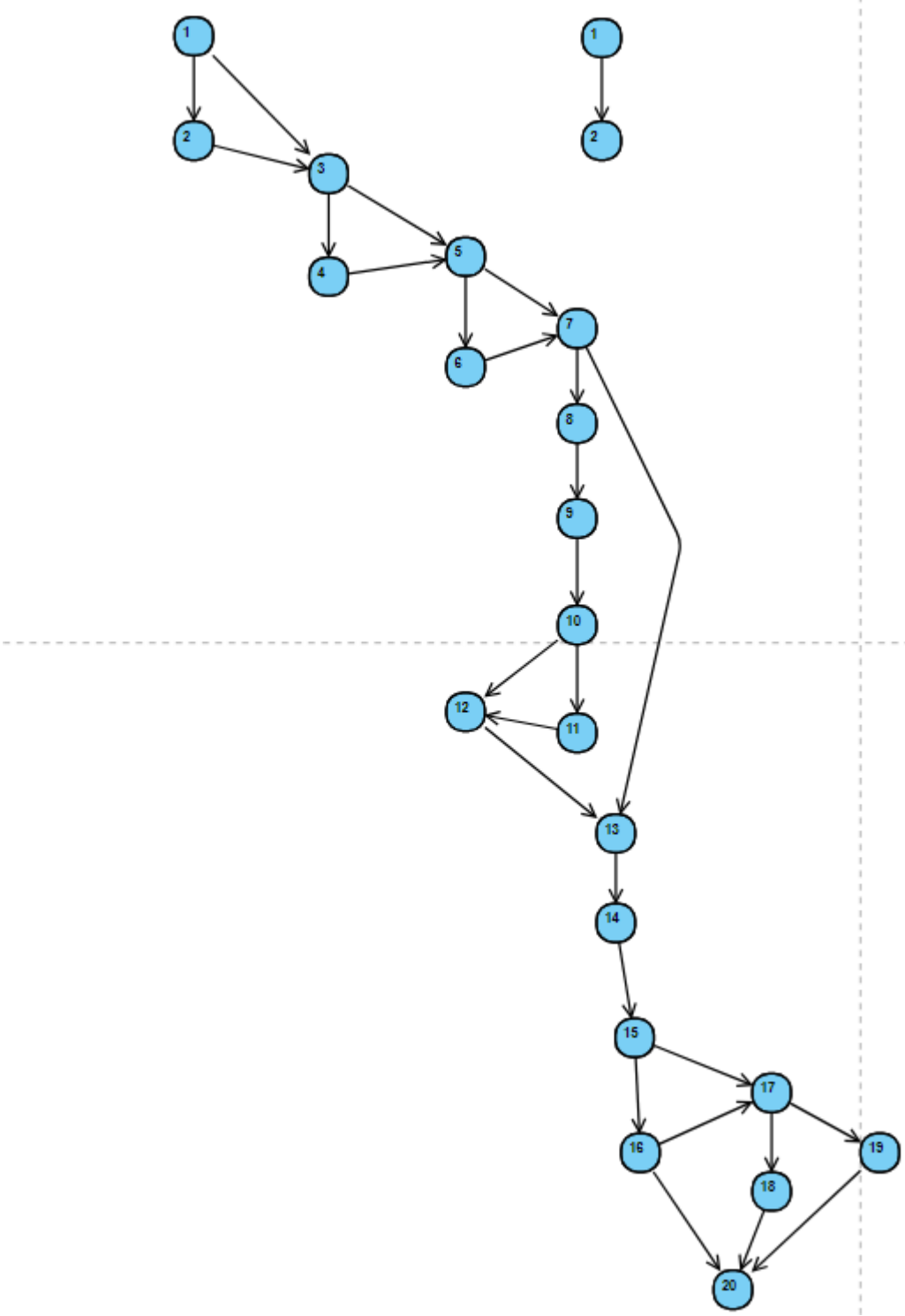
Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M2	Έχει εγκατασταθεί επιτυχώς το Ημερολόγιο και γίνεται Έλεγχος για σημερινή Κλήση	<code>hasMeeting == true</code>	<code>c.setToolTipText.(Meeting today, click for details)</code>
M3	Έχει εγκατασταθεί επιτυχώς το Ημερολόγιο και γίνεται Έλεγχος για σημερινές	<code>hasAnnouncement == true</code>	<code>c.setToolTipText.(Announcement today, click for details)</code>

	Ανακοινώσεις		
M4	Έχει εγκατασταθεί επιτυχώς το Ημερολόγιο και γίνεται Έλεγχος για Εργασίες	<code>hasProject == true</code>	<code>c.setToolTipText.(Project Deadline today, click for details)</code>

**Οι Γράφοι για τα listeners on 1 & 2 clicks**

calendarTable.addMouseListener 1 click

calendarTable.addMouseListener 2 clicks



calendarTable.listener for click 1

Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 27 - 20 + 2 = 9$  περιοχές,

όπως φαίνεται και στον γράφο.

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

M1: 1-3-5-7-13-14-15-16-20

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

M2: 1-2-3-7-13-14-15-17-19-20 (**steps for Main Flow**)

M3: 1-2-3-5-6-7-13-20, **σχεδιαστική παράλειψη έστω ότι υπάρχει ακμή από το 13-20** οπότε και οι περιοχές γίνονται 10 (**Alternative Flow 1**)

M4: 1-2-3-4-5-7-13-20 **σχεδιαστική παράλειψη έστω ότι υπάρχει ακμή από το 13-20** (**Alternative Flow 2**)

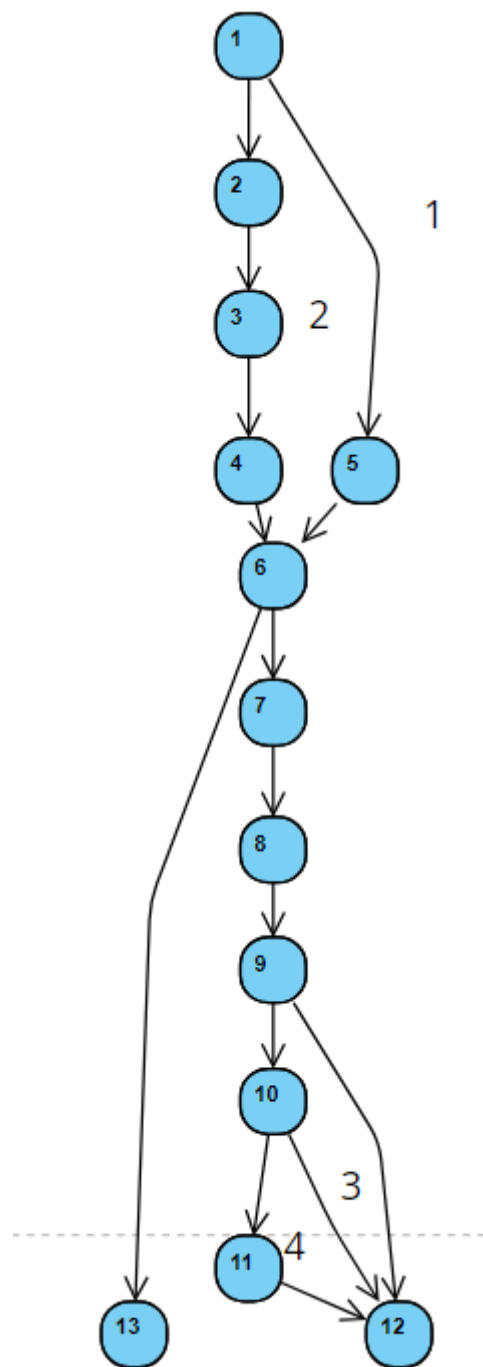
M5: 1-2-3-5-7-8-9-10-11-12-13-20 **σχεδιαστική παράλειψη έστω ότι υπάρχει ακμή από το 13-20** (**Alternative Flow 3**)

calendarTable.listener for click 2 (**Alternative flow 3**)

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα ('Εξοδος Προγράμματος')
M2	Κάνει κλικ 1 φορά πάνω στην σημερινή ημέρα. Ανάκτηση των σημερινών	detailsMeetings.containsKey(dayOf WeekFull) == true ("canceled".equalsIgnoreCase(status)) == false ("postponed".equalsIgnoreCase(status)) == false	Message: Your lecture for 'Subject' is 'status'

	ν πληροφο ριών		
M3	Κάνει κλικ 1 φορά πάνω στην σημερινή ημέρα. Ανάκτησ η των σημερινώ ν πληροφο ριών	detailsAnnouncement.containsKey( dayOfWeekFull) == true	Message: Admin says the system will be updated at..
M4	Κάνει κλικ 1 φορά πάνω στην σημερινή ημέρα. Ανάκτησ η των σημερινώ ν πληροφο ριών	detailsProject.containsKey(dayOfW eekFull) == true	Message: Today is the Deadline for Project in 'Subject'

Ο Γράφος για τη συνάρτηση loadUserSubjectMeetings:



Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 15 - 13 + 2 = 4$  περιοχές,

όπως φαίνεται και στον γράφο.

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

M1: 1-5-6-13

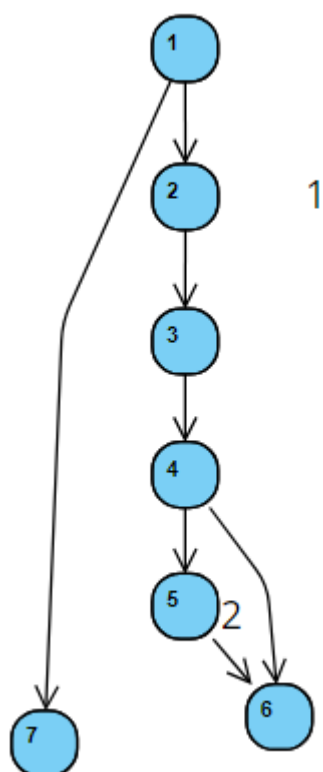
Ακολουθούν τα υπόλοιπα μονοπάτια που προκύπτουν με προσθήκη ακμών στο M1.

M2: 1-2-3-4-6-7-8-9-10-11-12 (**steps of Main flow**)

<b>Path</b>	<b>Περιγραφή</b>	<b>Περίπτωση Ελέγχου</b>	<b>Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)</b>
M1	Έλεγχος σύνδεσης στη Βάση	Both connections fail	e.printStackTrace()
M2	Κάνει κλικ 1 φορά πάνω στην σημερινή ημέρα. Ανάκτηση των σημερινών πληροφοριών	rsSubject.next() rsMeeting.next() (!meetingDetails.containsKey(dayOfWeek))	userSubjects.add(rsSubject.getString("subject_name")); meetingDetails.get(dayOfWeek).add(new String[] {meetingsAbout, host, startTime, endTime, status});

**Ο Γράφος για τη συνάρτηση loadUserReminders**





Σύμφωνα με τον τύπο:  $V(g) = e - n + 2 =$   
 $= 7 - 7 + 2 = 2$  περιοχές, όπως

φαίνεται και στον γράφο.

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

M1: 1-7

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

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

Path	Περιγραφή	Περίπτωση Ελέγχου	Αναμενόμενο Αποτέλεσμα (Έξοδος Προγράμματος)
M1	Έλεγχος σύνδεσης στη Βάση	connection fails	e.printStackTrace()

M 2	Ανακτού νται οι πληροφο ρίες για τις Υπενθυμ ίσεις και τοποθετ ούνται στη userRem inders.	rs.next() (!userReminders.contain sKey(reminderDate	userReminders.put(remi nderDate, new ArrayList<>());  userReminders.get(remi nderDate).add(new String[] {title, text})
--------	--	---	--