
APE-learn Platform



Sequence Diagrams-v1.0

Η Ομάδας μας και η Συμβολή της στο παρόν Τεχνικό Κείμενο

Μέλος	ΑΜ	Ρόλος
Πεντεσκούφης Παναγιώτης	1067470	Συντάκτης (Editor)
Πραμαντιώτης Χρήστος	1067457	Συνεργάτης (Contributor)
Αποστολόπουλος Γεώργιος	1067456	Συνεργάτης (Contributor)
Τσιρίκος Αναστάσιος	1070926	Συνεργάτης (Contributor)

Παραδοχές όσον αφορά την σχεδίαση των Διαγραμμάτων

- Τα Box Lifelines με μπλε χρώμα υποδηλώνουν κάποια Οθόνη
- Τα Box Lifelines με γκρι χρώμα υποδηλώνουν κάποια Οντότητα
- Τα Box Lifelines με κίτρινο χρώμα υποδηλώνουν κάποιο Use Case στο οποίο μεταφέρεται η ροή

Σε περίπτωση που κάποιο από τα Διαγράμματα Ακολουθίας δεν αναπαρίστανται με ευκρίνεια μπορείτε να μεταβείτε στο Προφίλ του APE-learn στο GitHub (στο Branch 'Sequence Diagrams'):

GitHub Link: <https://github.com/prChristos/APE-learn/tree/Sequence-Diagrams>

```

sequenceDiagram
    participant User
    participant System
    participant Message
    participant Message
    participant Message
    participant Student
    participant Regulator
    participant Request

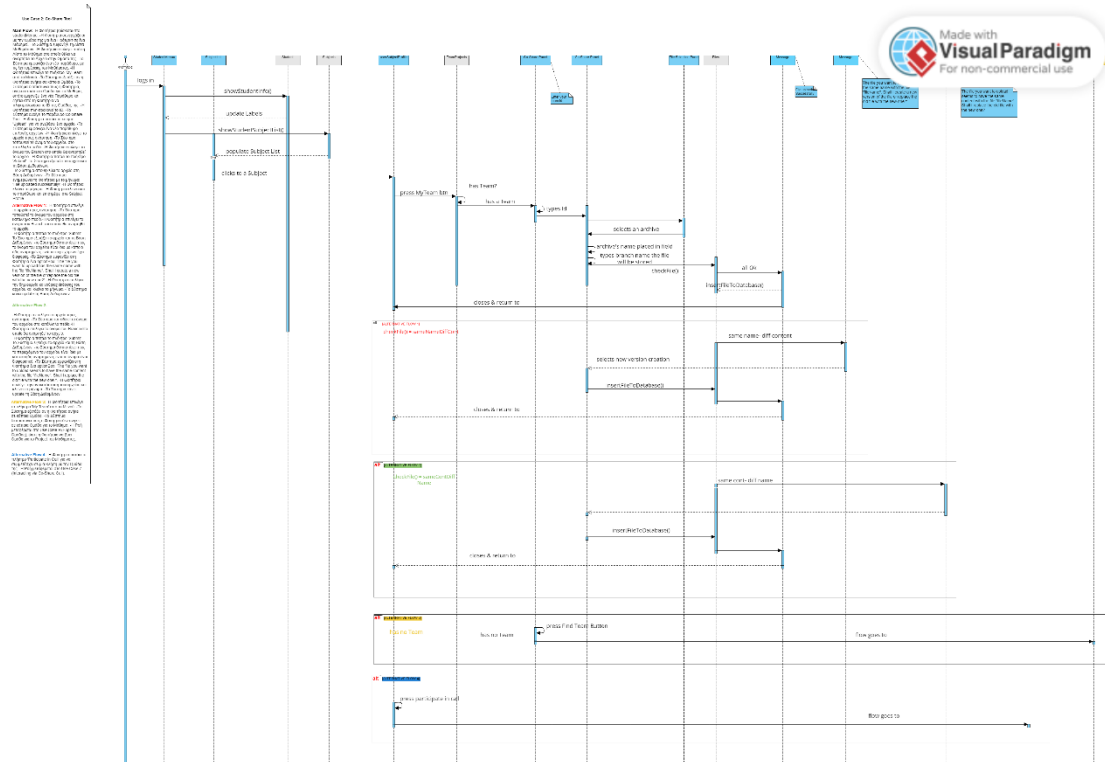
    User->>System: logs in
    System->>Message: showStudentInfo()
    Message->>System: populate Labels
    System->>Message: showBellIcon()
    Message->>System: no Notifications
    Message->>System: press the Bell
    System->>Message: checkNotifications()
    Message->>System: retrieveRegulationsDates()
    System->>Regulator: checkRequests()
    Regulator-->>System: no Notifications for Regulations and Requests
    System->>Message: showMessage()
    Message->>System: closes Message
    Message->>System: no Notifications

    alt [ALTERNATIVE FLOW 1]
        Message->>System: has Notifications
        Message->>System: press the Bell
        System->>Message: checkNotifications()
        Message->>System: retrieveRegulationsDates()
        System->>Regulator: checkRequests()
        Regulator-->>System: Notifications for Regulations
        Regulator-->>System: no Notifications for Requests
        System->>Message: showMessage()
        Message->>System: closes Message
        Message->>System: no Notifications
    end

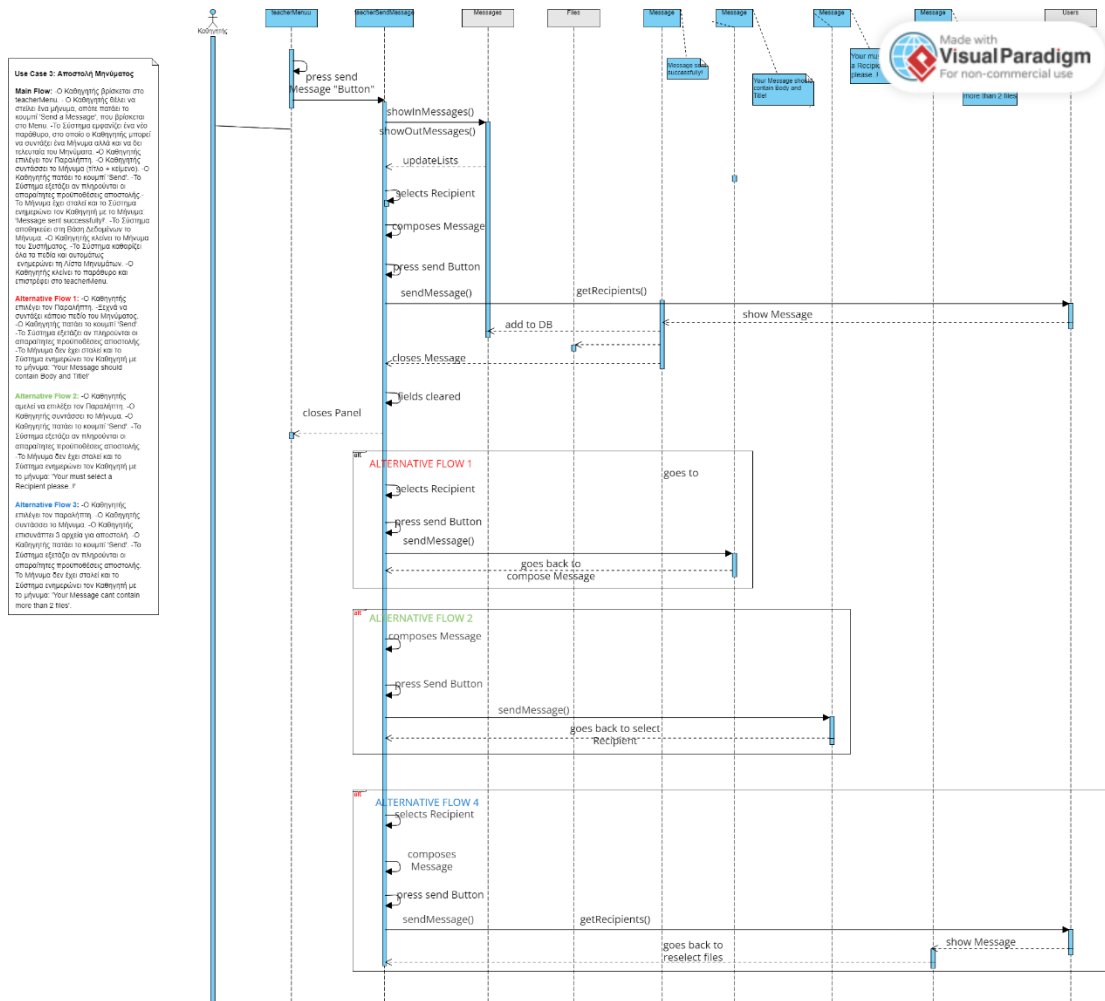
    alt [ALTERNATIVE FLOW 2]
        Message->>System: has Notifications
        Message->>System: press the Bell
        System->>Message: checkNotifications()
        Message->>System: retrieveRegulationsDates()
        System->>Regulator: checkRequests()
        Regulator-->>System: Notifications for Regulations
        Regulator-->>System: no Notifications for Requests
        System->>Message: showMessage()
        Message->>System: closes Message
        Message->>System: no Notifications
    end

    alt [ALTERNATIVE FLOW 3]
        Message->>System: has Notifications
        Message->>System: press the Bell
        System->>Message: checkNotifications()
        Message->>System: retrieveRegulationsDates()
        System->>Regulator: checkRequests()
        Regulator-->>System: Notifications for Requests
        Regulator-->>System: no Notifications for Regulations
        System->>Message: showMessage()
        Message->>System: closes Message
        Message->>System: no Notifications
    end
  
```

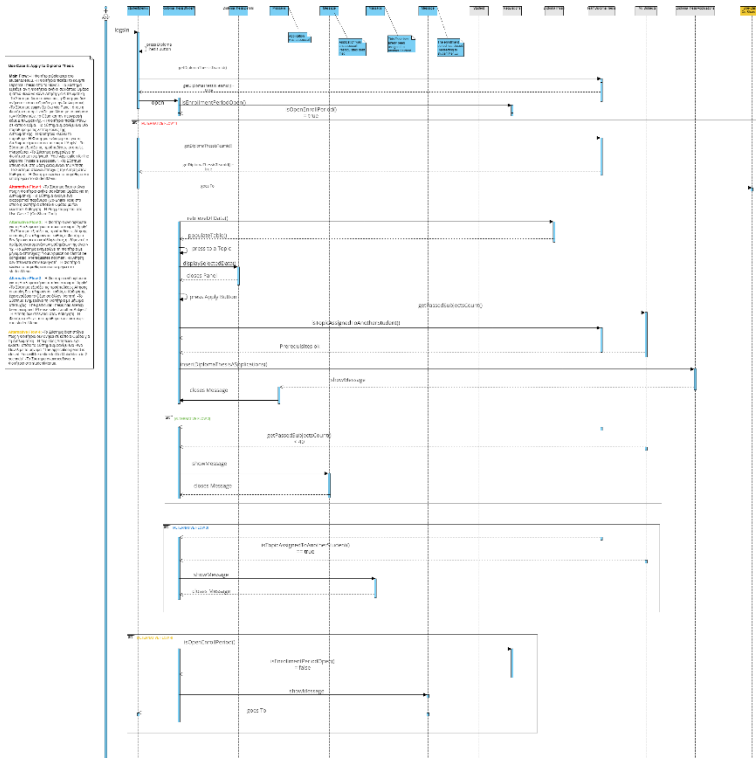
Sequence Diagram for Use Case 2



Sequence Diagram for Use Case 3



Sequence Diagram for Use Case 5

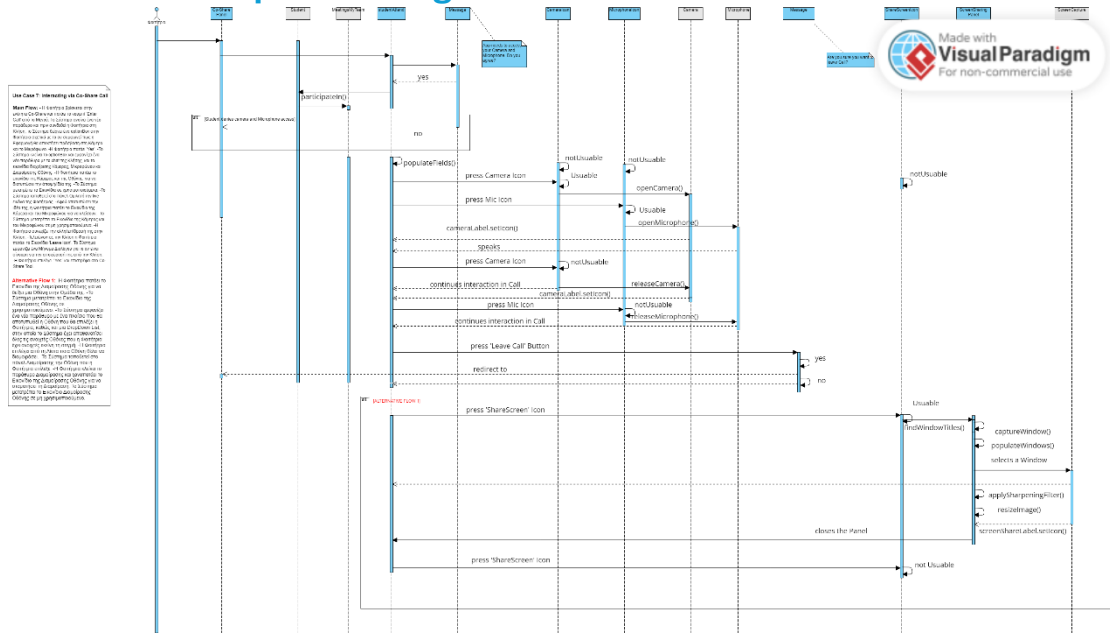


```

sequenceDiagram
    participant user
    participant server
    participant database
    participant lecture
    participant teacher
    participant student
    participant subject
    participant lobby
    participant cancel

    user->>server: login()
    activate server
    server->>database: retrieveStudentInfo()
    deactivate database
    server->>server: updateLabels()
    server->>server: showStudentSubjects()
    server->>server: populateList()
    server->>server: press to a Subject
    server->>server: show
    server->>server: press Attend to Lecture Button
    server->>server: checkStartHour()
    server->>server: isConnected()
    server->>server: notifyTeacher()
    server->>server: teacher accepts student
    server->>server: Lecture ends
    server->>server: redirect to
    server->>server: checkStartHour = more than 30 minutes
    server->>server: redirect to
    server->>server: checkStartHour = less than 30 minutes
    server->>server: add user to lobby
    server->>server: student enters lobby
    server->>server: remove student from lobby
    server->>server: goes to
    server->>server: has cancelled
    deactivate server
  
```


Sequence Diagram for Use Case 7



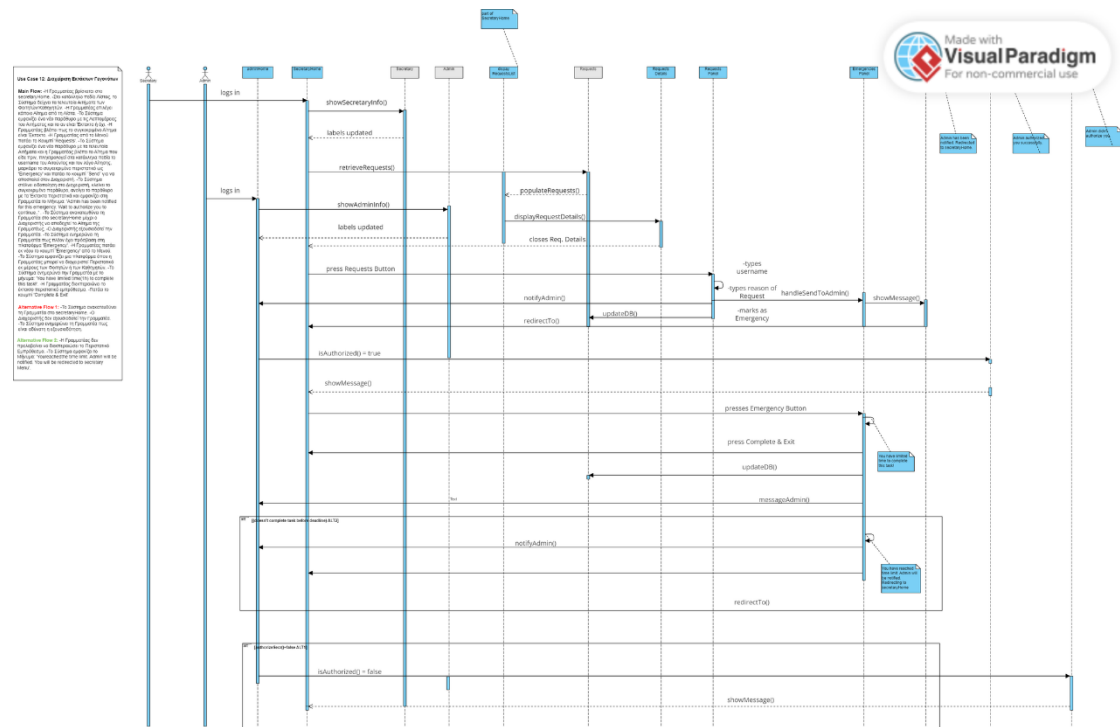
```

sequenceDiagram
    participant User
    participant Calendar
    participant Meeting
    participant Announcement
    participant Reminder

    User->>Calendar: login
    activate Calendar
    Calendar->>Calendar: setUpCalendar()
    deactivate Calendar
    Calendar->>Calendar: drawCalendar()
    activate Calendar
    Calendar->>Calendar: has a black dot today
    deactivate Calendar
    Calendar->>Calendar: clicks today's day
    activate Calendar
    Calendar->>Meeting: loadUserSubjectMeetings()
    activate Meeting
    Meeting->>Calendar: detailsMeetings
    deactivate Meeting
    Calendar->>Calendar: showMessage
    deactivate Calendar
    Calendar->>Calendar: has a green dot today
    deactivate Calendar
    Calendar->>Calendar: clicks today's day
    activate Calendar
    Calendar->>Announcement: loadAnnouncementDates()
    activate Announcement
    Announcement->>Calendar: announcementDetails
    deactivate Announcement
    Calendar->>Calendar: showMessage
    deactivate Calendar
    Calendar->>Calendar: closes Message
    deactivate Calendar
    Calendar->>Calendar: system maintain the dot
    deactivate Calendar
    Calendar->>Calendar: has a blue dot today
    deactivate Calendar
    Calendar->>Calendar: clicks today's day
    activate Calendar
    Calendar->>Reminder: loadProjectDates()
    activate Reminder
    Reminder->>Calendar: projectDetails
    deactivate Reminder
    Calendar->>Calendar: showMessage
    deactivate Calendar
    Calendar->>Calendar: closes Message
    deactivate Calendar
    Calendar->>Calendar: system maintain the dot
    deactivate Calendar
    Calendar->>Calendar: double click on a day
    activate Calendar
    Calendar->>Reminder: handleReminderInput
    activate Reminder
    Reminder->>Reminder: types Reminder
    deactivate Reminder
    Reminder->>Reminder: clicks on set button
    deactivate Reminder
    Calendar->>Reminder: insertReminderIntoDatabase()
    activate Reminder
    Reminder->>Calendar: black dot placed in cell Day
    deactivate Reminder
    Calendar->>Calendar: closes Message
    deactivate Calendar
    Calendar->>Calendar: showMessage
    deactivate Calendar
    
```

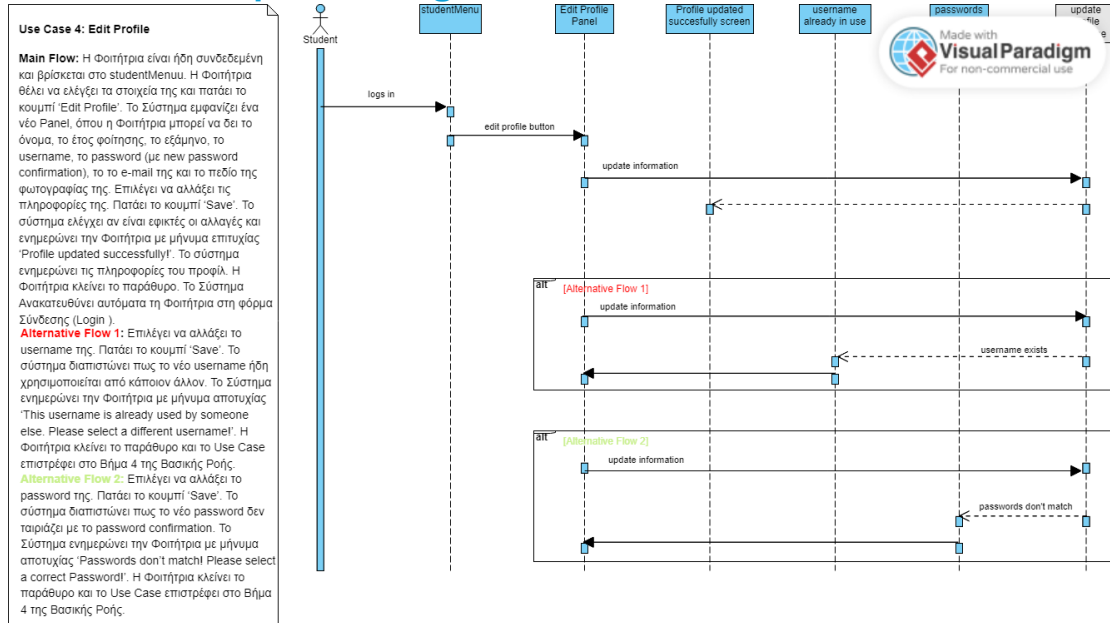
The diagram illustrates the sequence of interactions for a calendar application. It starts with a user logging in, followed by the calendar being set up and drawn. The user interacts with the calendar by clicking on a day, which triggers the loading of meetings and announcements. The user then views details and closes messages. The process continues with the user clicking on a day again, leading to the loading of project dates and the insertion of a reminder into the database. Finally, the user handles the reminder input, types the reminder, and clicks on the set button, which results in a black dot being placed in the calendar cell.

Sequence Diagram for Use Case 12

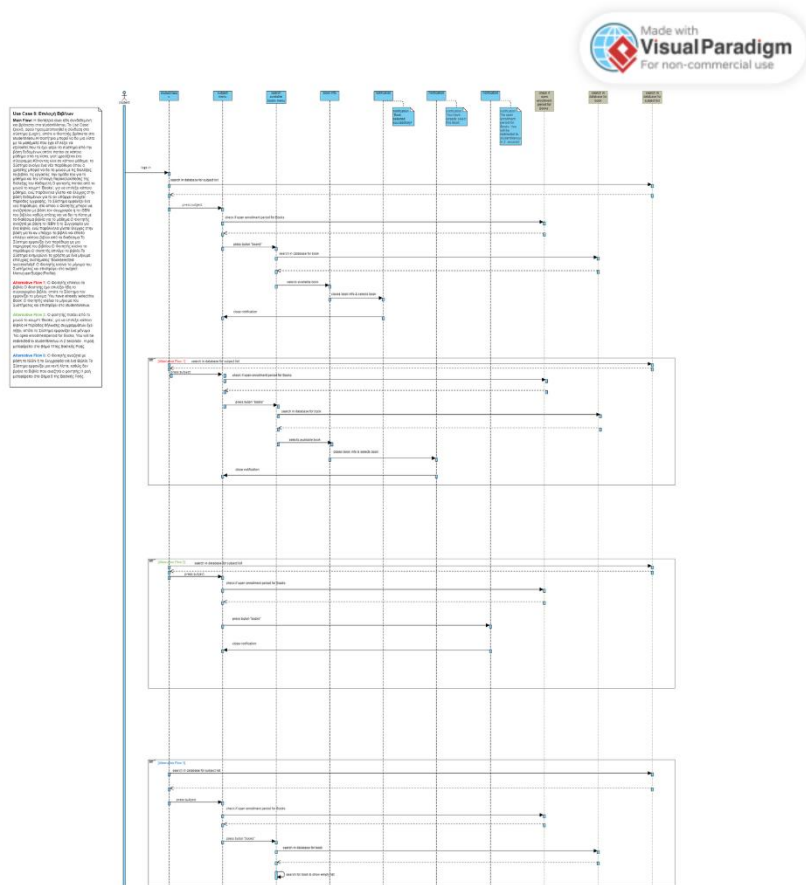


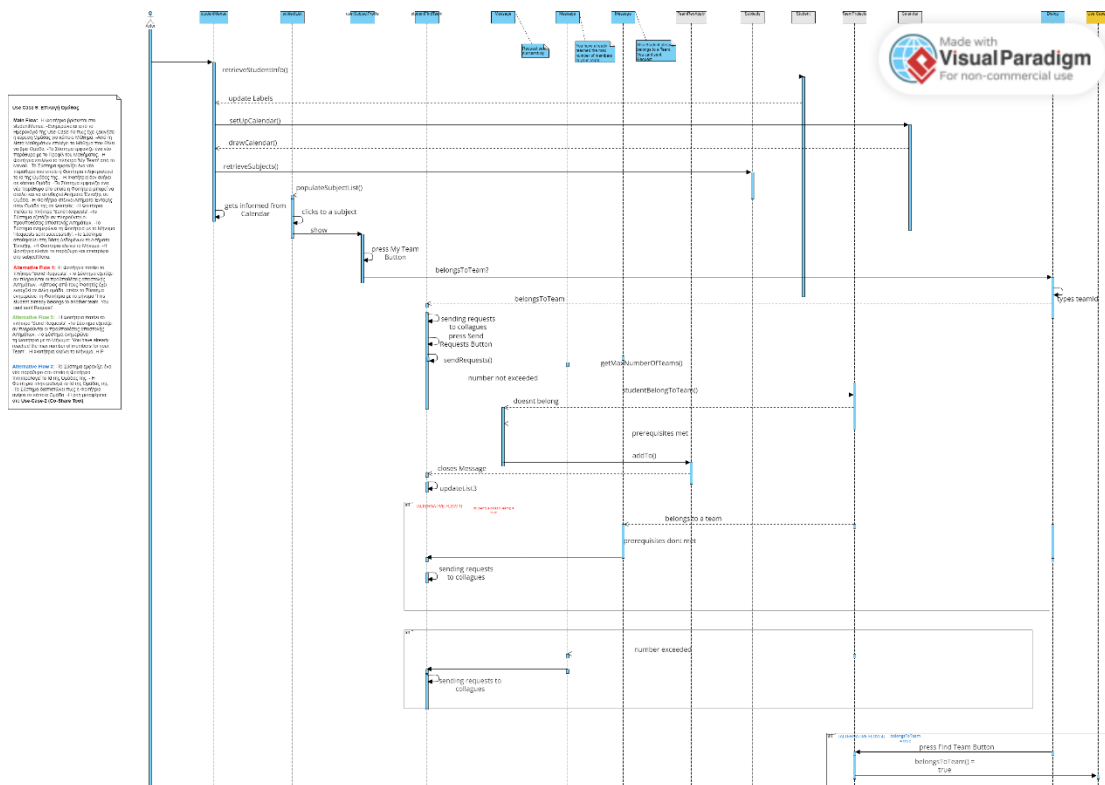
Sequence Diagram for the secondary Use Cases

Sequence Diagram for Use Case 4

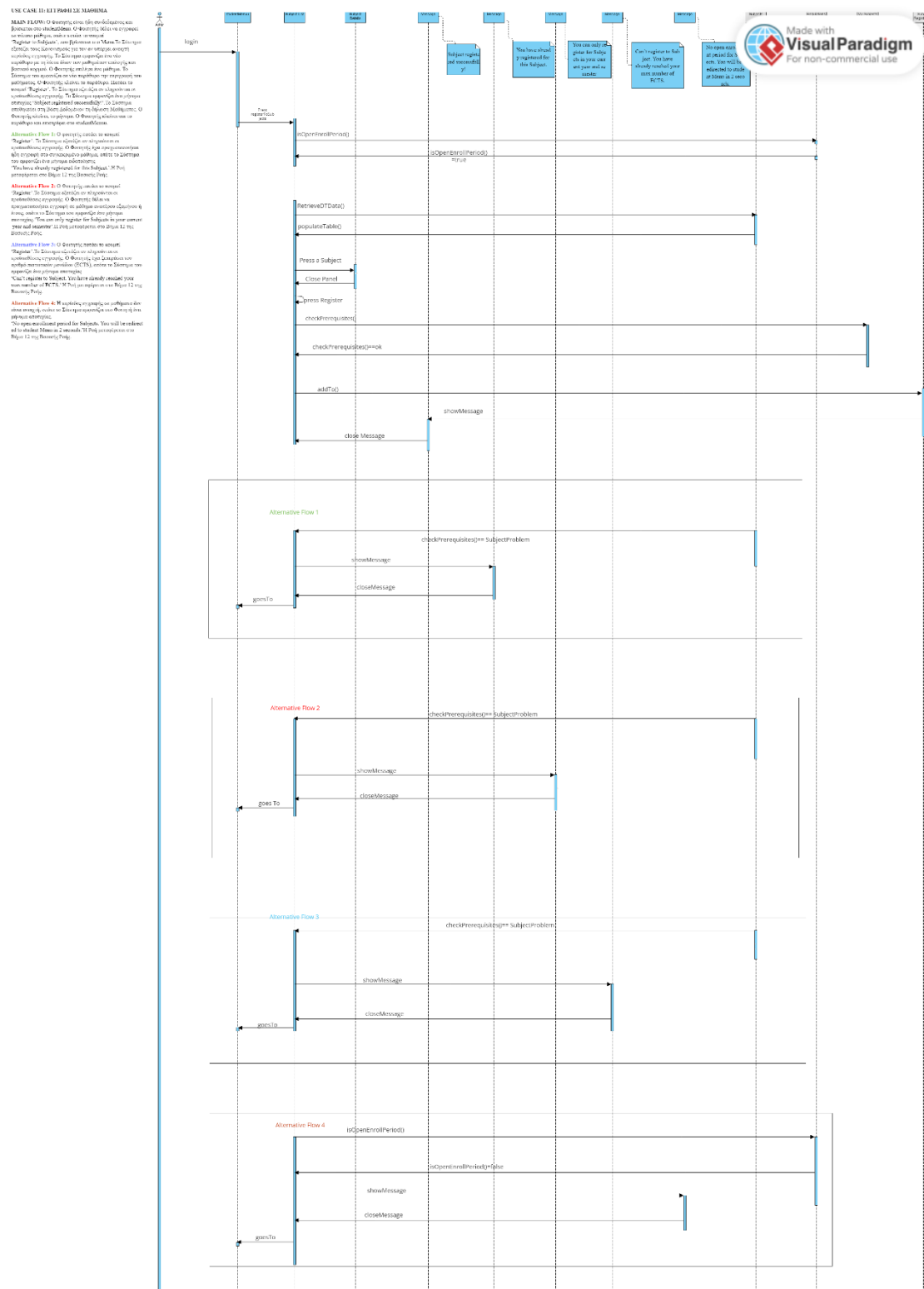


Sequence Diagram for Use Case 8





Sequence Diagram for Use Case 11



Εργαλεία που χρησιμοποιήθηκαν στο Παρόν Παραδοτέο

- Για τη συγγραφή χρησιμοποιήσαμε το Microsoft Word.
- Για τη σχεδίαση των Διαγραμμάτων Ακολουθίας χρησιμοποιήσαμε το Visual Paradigm.