



**Εργασία μαθήματος**

**<<ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ>>**

Θέμα εργασίας: 4

**Αναβολή στράτευσης**

(Ανάλυση, μοντελοποίηση και υλοποίηση)

Ομάδα: 67

Ευστράτιος Ορφανός

Παύλος Ανδρέου

ΑΜ: 21875

ΑΜ: 218110

Επιβλέποντες καθηγητές:

Δρ. Μάρα Νικολαΐδου

Δρ. Ανάργυρος Τσαδήμας

**Ταύρος, Ιανουάριος 2022**

Πίνακας Περιεχομένων

[**1.** **Εισαγωγή** 3](#_Toc91525654)

[1.1 Στόχος εργασίας 3](#_Toc91525655)

[1.2 Στάδια έργου 3](#_Toc91525656)

[1.3 Εργαλεία που χρησιμοποιήθηκαν 5](#_Toc91525657)

[1.4 Περιγραφή θέματος εργασίας 5](#_Toc91525658)

[1.5 Ορισμοί – Ρόλοι 6](#_Toc91525659)

[**2.** **Τεύχος σχεδιασμού** 8](#_Toc91525660)

[2.1. Λειτουργικές απαιτήσεις 8](#_Toc91525661)

[2.2. Παραδοχές - υποθέσεις 8](#_Toc91525662)

[2.3 Αρχιτεκτονική Συστήματος 11](#_Toc91525663)

[2.4 ER διάγραμμα 13](#_Toc91525664)

[2.5 Use Case διαγράμματα 14](#_Toc91525665)

[2.5.1 High Level Use Case Diagram 14](#_Toc91525666)

[2.5.2 Use Case για την Διαχείριση αιτήσεων αναβολής στράτευσης 15](#_Toc91525667)

[2.5.3 Use Case για την Διαχείριση στρατολογικών γραφείων 16](#_Toc91525668)

[2.5.4 Use Case για την Διαχείριση καταλόγου χρηστών 17](#_Toc91525669)

[2.6 Class διάγραμμα 18](#_Toc91525670)

[2.7 Sequence διαγράμματα 22](#_Toc91525671)

[2.7.1 Πολίτης 22](#_Toc91525672)

[Νέα αίτηση αναβολής στράτευσης 22](#_Toc91525673)

[2.7.2 Αξιωματικός 26](#_Toc91525674)

[2.7.3 Υπάλληλος 27](#_Toc91525675)

[2.7.4 Διαχειριστής Συστήματος 27](#_Toc91525676)

[2.8 State machine διαγράμματα 35](#_Toc91525677)

[2.8.1 State machine διαγράμμα: Αναβολή στράτευσης 35](#_Toc91525678)

[2.9 Component διάγραμμα 37](#_Toc91525679)

[2.10 Deployment διάγραμμα 37](#_Toc91525680)

[3. Τεύχος υλοποίησης της εφαρμογής 37](#_Toc91525681)

[4. Εγχειρίδιο χρήσης 37](#_Toc91525682)

## **Εισαγωγή**

### 1.1 Στόχος εργασίας

Στα πλαίσια του μαθήματος Κατανεμημένα Συστήματα κληθήκαμε να αναλύσουμε, μοντελοποιήσουμε και να υλοποιήσουμε ένα σύστημα διαχείρησης αιτήσεων για αναβολή στράτευσης των πολιτών. Στόχος είναι να κατανοήσουμε τον τρόπο με τον οποίο υλοποιούνται οι σύγχρονες διαδικτυακές εφαρμογές, τις αρχιτεκτονικές που έχουν καθιερωθεί, καθώς και τις βασικές τεχνολογίες πού χρησιμοποιούνται για την κατασκευή τέτοιων συστημάτων.

### 1.2 Στάδια έργου

Η εργασία θα ολοκληρωθεί σε τέσσερα στάδια.

Το πρώτο στάδιο περιλαμβάνει την ανάλυση και μοντελοποίηση του έργου. Λάβαμε υπόψιν μας και αναλύσαμε όλα τα δεδομένα που μας δόθηκαν, κατανοήσαμε το θέμα, καθορίσαμε τις λειτουργικές απαιτήσεις, διασαφηνίσαμε τις παραδοχές-υποθέσεις και έπειτα προβήκαμε στη μοντελοποίηση με διαγράμματα με την χρήση της UML για να αποτυπώσουμε τις ιδέες μας με ακρίβεια.

Το δεύτερο στάδιο περιλαμβάνει την υλοποίηση του εσωτερικού συστήματος (intranet), καθώς και του υποσυστήματος της βάσης δεδομένων όπου θα διατηρούνται τα στοιχεία της εφαρμογής. Εν συντομία οι εργασίες που θα γίνουν θα αφορούν την παραμετροποίηση του Application Server της εφαρμογής, την υλοποίηση του Controller, τον σχεδιασμό της βάσης δεδομένων, την σύνδεση της εφαρμογής με τη βάση δεδομένων, το σχεδιασμό των οθονών των εσωτερικών χρηστών και το business logic της εφαρμογής. Η ανάπτυξη θα γίνει με τεχνολογία Web Services που θα τρέχουν στον Controller. Τα Web Services αναπτυχθούν με το πακέτο Spring framework ενώ για τη σύνδεση της εφαρμογής με τη βάση δεδομένων θα χρησιμοποιηθεί η τεχνολογία hibernate.

Το τρίτο στάδιο περιλαμβάνει την υλοποίηση του εξωτερικού συστήματος της εφαρμογής (internet). Ουσιαστικά αυτό που βλέπουν οι πολίτες. Η ανάπτυξη της τρίτης φάσης θα γίνει με τεχνολογία RESTful API. Στο backend θα χρησιμοποιηθεί ένας web server ο οποίος θα ανακατευθύνει τα αιτήματα στον Application Server. Έτσι δεν επιτρέπουμε τους εξωτερικούς χρήστες να έχουν άμεση πρόσβαση στον Application Server. Αυτό γίνεται για λόγους ασφαλείας.

Το τέταρτο στάδιο περιλαμβάνει την τεκμηρίωση (documentation) της εφαρμογής, την συγγραφή των οδηγιών χρήσης καθώς και την παρουσίαση της εφαρμογής.

### 1.3 Εργαλεία που χρησιμοποιήθηκαν

Τα εργαλεία που χρησιμοποιήθηκαν είναι:

* Visual Paradigm, όπου σε αυτό υλοποιήσαμε σχεδόν όλα τα διαγράμματα που παρατίθενται στην συνέχεια. Συγκεκριμένα τα Use Case diagrams, Class diagram, Sequence diagrams, State Machine diagrams και Deployment diagram.
* MySQL Workbench για το Entity Relationship (ER) diagram.
* Jet Brains IntelliJ IDEA Ultimate ως βασικό εργαλείο προγραμματισμού της εφαρμογής.
* Apache Tomcat version 9 για Application Server.
* mySQL Database για Βάση Δεδομένων.

Τεχνολογίες Υλοποίησης

* Spring Boot + Spring Data JPA για τη υλοποίηση του Backend
* Spring security για τη δημιουργία ρόλων και χρηστών
* Angular για Frontend

### 1.4 Περιγραφή θέματος εργασίας

Το θέμα είναι διαχρονικό και έχει να κάνει με την διαχείριση των αιτήσεων για αναβολή στράτευσης των πολιτών που αποστέλλονται στα στρατολογικά γραφεία. Η γενική περιγραφή του θέματος είναι:

«Ένα σύστημα όπου ένας πολίτης θα μπορεί να αιτηθεί αναβολή στράτευσης για κάποιους λόγους (σπουδές, υγείας κτλ). Στην αίτηση θα πρέπει να υποβάλλονται τα στοιχεία του πολίτη, ο στρατολογικός αριθμός του και κάποιο έγγραφο που να τεκμηριώνει το λόγο της αναβολής. Την αίτηση θα επικυρώνει ένας υπάλληλος του στρατολογικού γραφείου που ανήκει ο πολίτης και θα την εγκρίνει ένας αξιωματικός του στρατολογικού γραφείου.»

Στόχος της εφαρμογής αυτής είναι οι πολίτες που θέλουν αναβολή στράτευσης να είναι πλέον ικανοί να διαχειρίζονται ευκολότερα, αποτελεσματικά και απομακρυσμένα τα αιτήματα τους για αναβολή στράτευσης με τα στρατολογικά γραφεία στα οποία ανήκουν.

### 1.5 Ορισμοί – Ρόλοι

Φορέας: Κάθε δημόσια υπηρεσία στην οποία απευθύνεται ο πολίτης για να λάβει εξυπηρέτηση σχετικά με θέματα που άπτονται της αρμοδιότητας της κάθε υπηρεσίας. Παράδειγμα: Δημόσια οικονομική υπηρεσία ΔΟΥ, ΙΚΑ, Δήμος, κλπ.

Προϊστάμενος φορέα: Είναι αυτός που θα κάνει την αίτηση για την συμμετοχή ενός φορέα στο σύστημα. Είναι υπεύθυνος, προφανώς, για τον δικό του φορέα και μόνο.

Υπάλληλος φορέα: Εργάζεται για έναν συγκεκριμένο φορέα και κύρια δουλειά του είναι να εγκρίνει ή να απορρίπτει τις αιτήσεις των πολιτών για ραντεβού με τον αντίστοιχο δημόσιο φορέα.

Διαχειριστής φορέων: Θα λάβει όλες τις αιτήσεις διάφορων προϊσταμένων και αφού τις επεξεργαστεί θα κρίνει αν πρέπει να εγκριθούν και να εισέλθουν στο σύστημα ή να απορριφθούν.

Διαχειριστής Συστήματος (Admin): Είναι ο χρήστης που κάνει την διαχείριση του Active Directory. Η δημιουργία νέων χρηστών, όπως επίσης και η τροποποίηση των στοιχείων τους ή η διαγραφή τους μπορεί να πραγματοποιηθεί μόνο από τον διαχειριστή του συστήματος. Επιπλέον είναι υπεύθυνος για την δημιουργία των ρόλων που μπορεί να έχουν οι χρήστες στην εφαρμογή καθώς και για τη διαχείρισή τους. Επίσης είναι υπεύθυνος για την εισαγωγή των νέων υπηρεσιών που προσφέρονται καθώς και για τα δικαιώματα πρέπει να έχει κάποιος για να του είναι διαθέσιμες.

# **Τεύχος σχεδιασμού**

## 2.1. Λειτουργικές απαιτήσεις

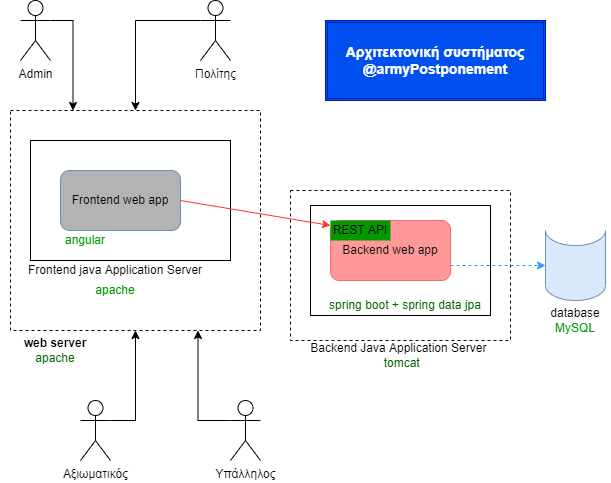
1. Οι βασικές λειτουργίες του συστήματος καθώς και οι βοηθητικές απεικονίζονται στα Use Case Διαγράμματα. Η περιγραφή των βασικών λειτουργιών γίνεται με τα Sequence Διαγράμματα κάτω από τα οποία ακολουθεί και μια σύντομη περιγραφή της λειτουργίας που περιγράφουν.
2. Ο Πολίτης θα πρέπει από μόνος του να μπορεί να εγγραφεί (Register) στο σύστημα με τον πιο απλό ρόλο αυτόν του πολίτη και στη συνέχεια να κάνει τις λειτουργίες που επιθυμεί.
3. Οι υπόλοιποι χρήστες εκτός από τον πολίτη θα πρέπει να τους κάνει εισαγωγή ο διαχειριστής του συστήματος Τους ανάλογους ρόλους, ο οποίος θα ενημερωθεί για το αίτημα της εισαγωγής με τρόπο εκτός της εφαρμογής, π.χ με e-mail ή με τηλέφωνο.
4. Ο διαχειριστής του συστήματος σε καμία περίπτωση δεν είναι ένας από τους χρήστες του συστήματος.
5. Ο πολίτης θα ενημερώνεται αυτόματα από το συστημα με email για το αποτέλεσμα της αίτησής του.

## 2.2. Παραδοχές - υποθέσεις

Έχοντας επεξεργαστεί το θέμα που καλούμαστε να υλοποιήσουμε χρειάστηκε να λάβουμε υπόψιν μας ορισμένες παραδοχές, οι οποίες είναι αναγκαίο να διασαφηνιστούν.

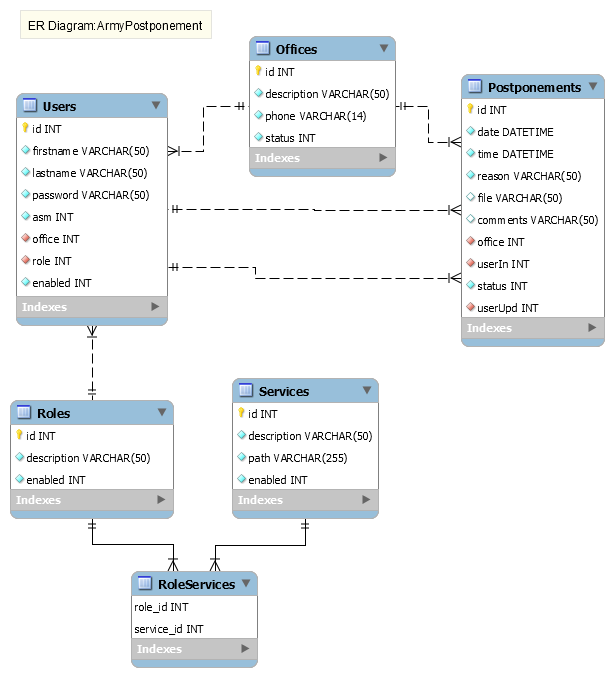
1. Για να υπάρξει πρόσβαση σε οποιαδήποτε από τις προσφερόμενες λειτουργίες του συστήματος θα πρέπει πρώτα να γίνει authentication του χρήστη με τη λειτουργία Login.
2. Για λόγους ασφαλείας οι πολίτες θεωρούνται εξωτερικό μέρος της εφαρμογής (internet) οι οποίοι για να εξυπηρετηθούν απευθύνονται στον Web Server ο οποίος είναι προσβάσιμος από το διαδίκτυο. Όλοι οι υπόλοιποι χρήστες με ρόλους διαφορετικούς του πολίτη χρησιμοποιούν το intranet του οργανισμού μας. Οι προϊστάμενοι φορέων καθώς και υπάλληλοι φορέων που παρόλο που είναι σε διαφορετικούς χώρους, τους θεωρούμε ως εσωτερικούς χρήστες οι οποίοι συνδέονται στο σύστημά μας μέσω της υπηρεσίας VPN.
3. Για το σύστημα μας ορίζουμε ως φορέα όχι το σύνολο του οργανισμού που προφέρει μια υπηρεσία, αλλά κάθε ένα από τα υποκαταστήματα του. Για να γίνει πιο κατανοητό ένας φορέας είναι η Δ.Ο.Υ Γλυφάδας και ένας διαφορετικός η Δ.Ο.Υ Χολαργού και δεν θεωρούμε την Δ.Ο.Υ έναν φορέα γενικά.
4. Όλοι οι υπάλληλοι ενός φορέα έχουν πρόσβαση σε όλα τα ραντεβού του φορέα τους. Επομένως η έγκριση και η απόρριψη των ραντεβού δεν διεκπεραιώνεται από έναν μόνο υπάλληλο, αλλά από όλους τους υπαλλήλους του ίδιου φορέα.
5. Από τη στιγμή που ένα αίτημα ενός Πολίτη για ραντεβού εγκριθεί ή απορριφθεί από τον υπάλληλο του φορέα, ο πολίτης δεν θα ειδοποιηθεί για την εξέλιξη του αιτήματος του. Για να δει αν το αίτημα που εγκρίθηκε η απορρίφθηκε θα πρέπει να ξαναμπεί στο σύστημα, να επιλέξει τη λειτουργία Νέο ραντεβού, όπου θα του εμφανιστούν όλα του τα ραντεβού και από κει θα δει αν έχει εγκριθεί ή απορριφθεί το αίτημα του. Στην περίπτωση που έχει απορριφθεί το αίτημα του από την ίδια οθόνη θα μπορεί να κάνει και ένα νέο αίτημα για ραντεβού.
6. Όσον αφορά την έγκριση ή την απόρριψη των φορέων από τον διαχειριστή φορέων σε καμία από τις δύο περιπτώσεις ο ενδιαφερόμενος φορέας δεν θα λάβει σχετική ειδοποίηση αφότου έχει επεξεργαστεί η αίτηση του από το σύστημα. Η ειδοποίηση θα πρέπει να γίνει με τρόπο εκτός της εφαρμογής π.χ. με e-mail ή με τηλέφωνο από τον διαχειριστή φορέων τους αιτούντα.
7. Στο υποσύστημα της εφαρμογής για την διαχείριση ρόλων έχει αναπτυχθεί μόνο η εισαγωγή και η διαγραφή ρόλων. Η ενημέρωση ρόλων δεν έχει αναπτυχθεί επομένως οι διορθώσεις δεν είναι εφικτές.
8. Το υποσύστημα της διαχείρισης των υπηρεσιών Που έχει σχεδιαστεί και απεικονίζεται στα διαγράμματα δεν θα υλοποιηθεί στην παρούσα έκδοση της εφαρμογής. Οι παρεχόμενες υπηρεσίες της εφαρμογής καθώς και οι ρόλοι που μπορούν να έχουν πρόσβαση σε αυτές, στην παρούσα έκδοση της εφαρμογής θα είναι hardcoded.

## 2.3 Αρχιτεκτονική Συστήματος



2.4 ER διάγραμμα

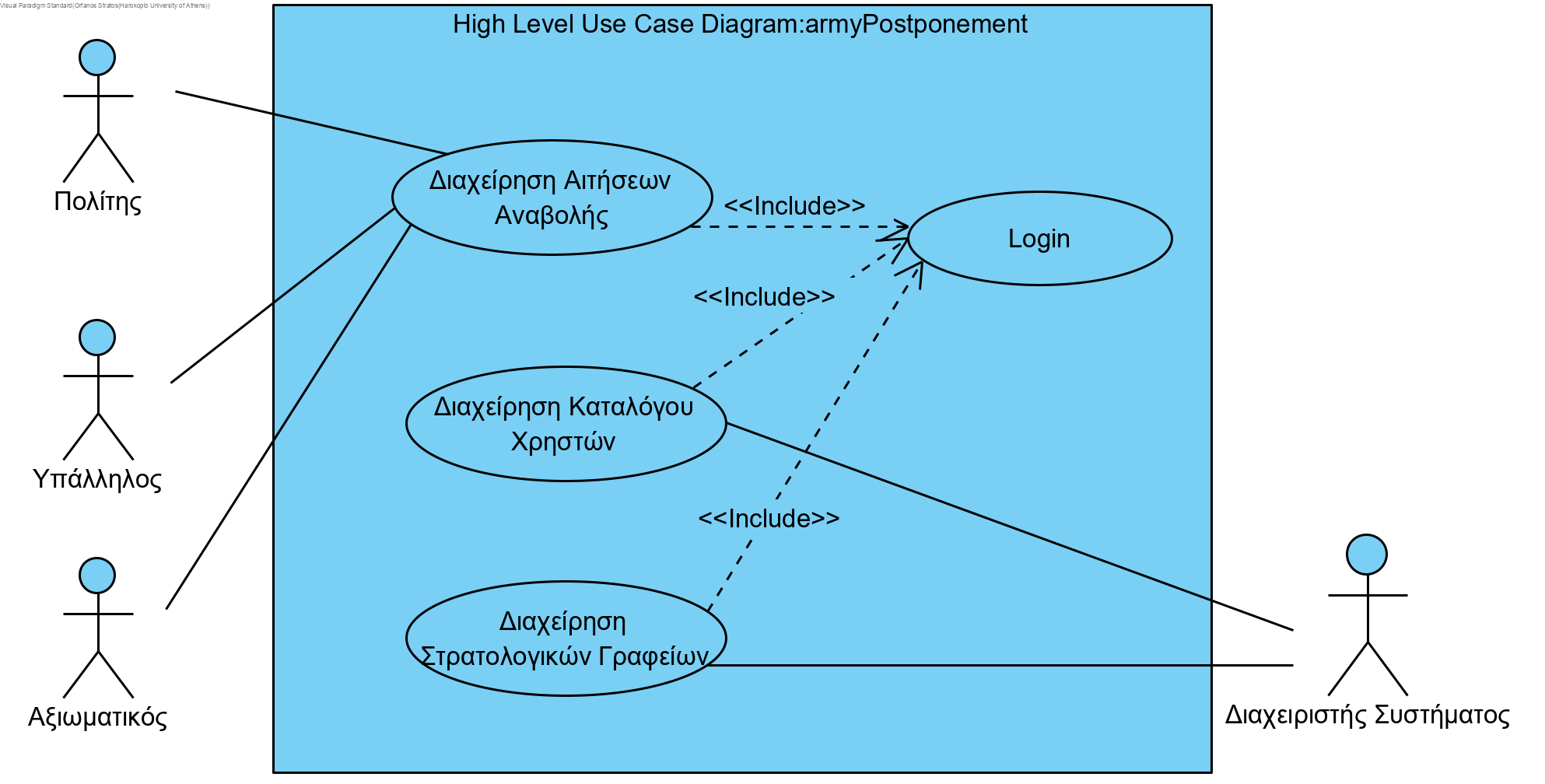
Παρακάτω ακολουθεί το ER diagram που απεικονίζει τις σχέσεις και τις εξαρτήσεις τους στην βάση δεδομένων.



## 2.5 Use Case διαγράμματα

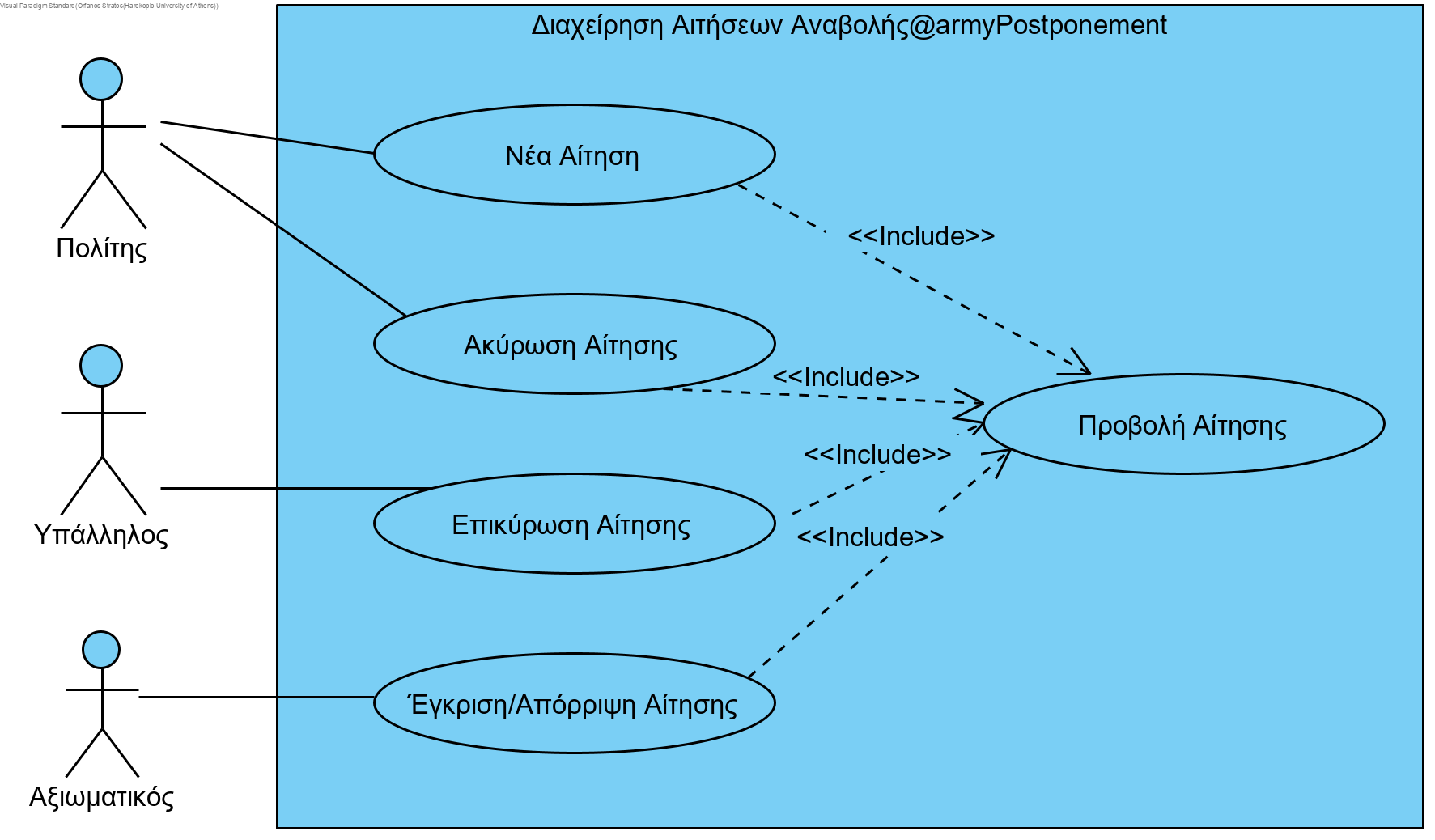
Σε πρώτη φάση θα δούμε όλα τα Use Case που υλοποιήσαμε. Θεωρήσαμε προτιμότερο να φτιάξουμε ένα High Level Use Case Diagram και έπειτα Use Case με πιο συγκεκριμένες λειτουργείες προκειμένου να αποτυπωθούν πιο κατανοητά οι λειτουργιες της εφαρμογής.

### 2.5.1 High Level Use Case Diagram



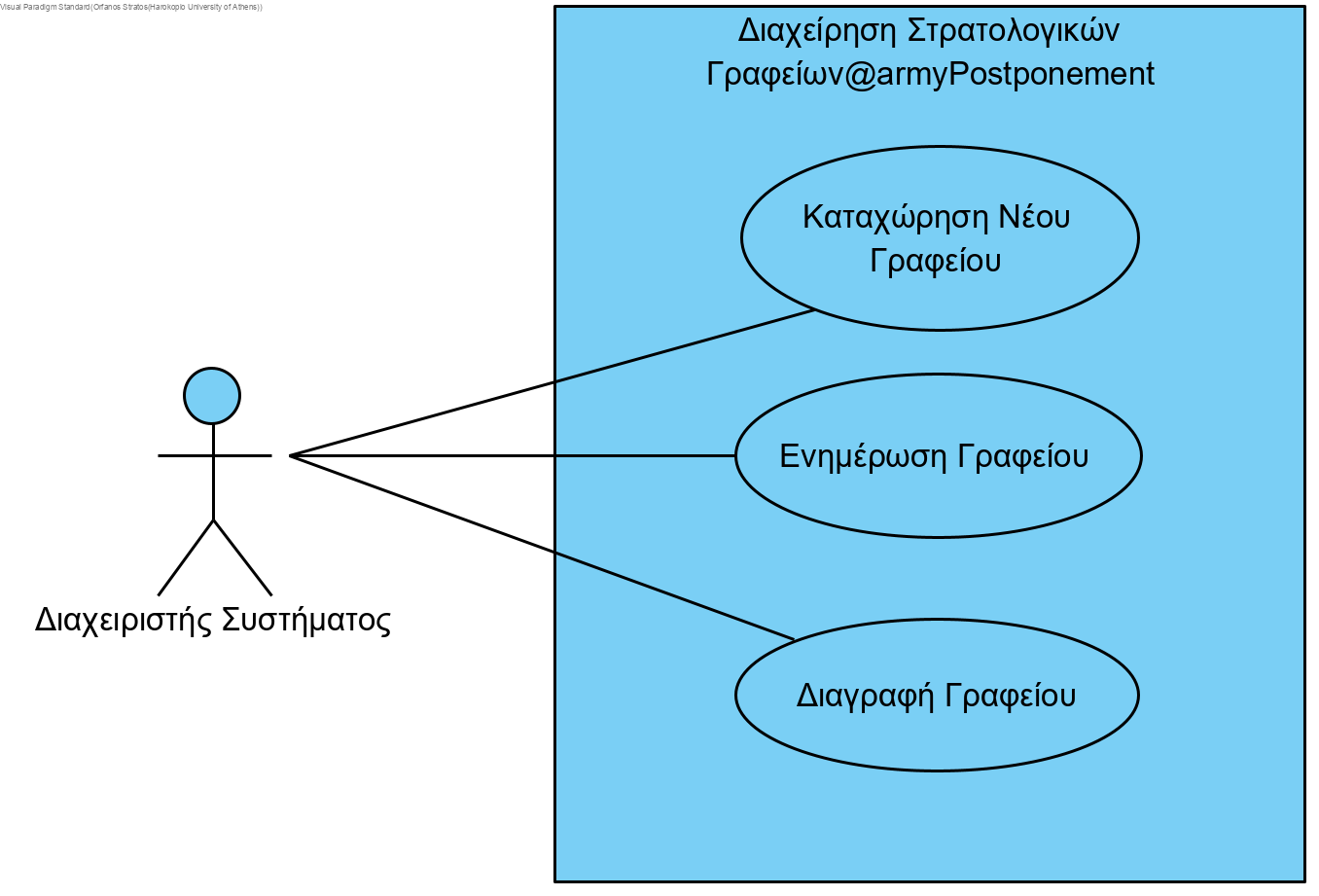
Στο παραπάνω διάγραμμα βλέπουμε σε γενικές γραμμές τις βασικές λειτουργίες του συστήματος καθώς και τους actor που σχετίζονται με την κάθε μια. Όπως φαίνεται το login είναι μια διαδικασία που περιλαμβάνεται σε όλες τις υπόλοιπες.

### 2.5.2 Use Case για την Διαχείριση αιτήσεων αναβολής στράτευσης



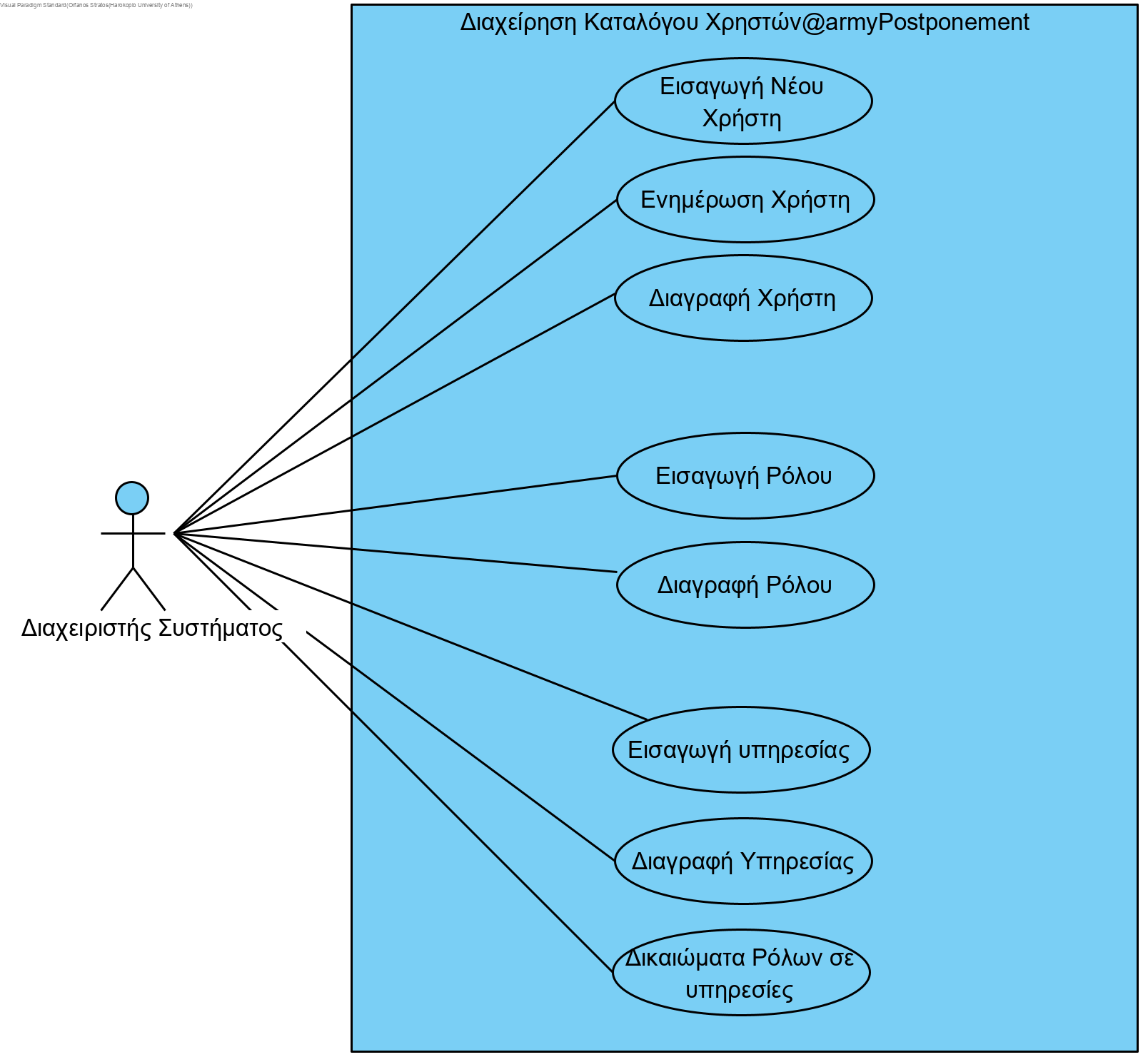
Στο παραπάνω διάγραμμα φαίνονται οι τρεις λειτουργίες που αντιστοιχούν στη διαχείριση ραντεβού. Η προβολή ραντεβού είναι μια διαδικασία που περιλαμβάνεται σε όλες τις υπόλοιπες.

### 2.5.3 Use Case για την Διαχείριση στρατολογικών γραφείων



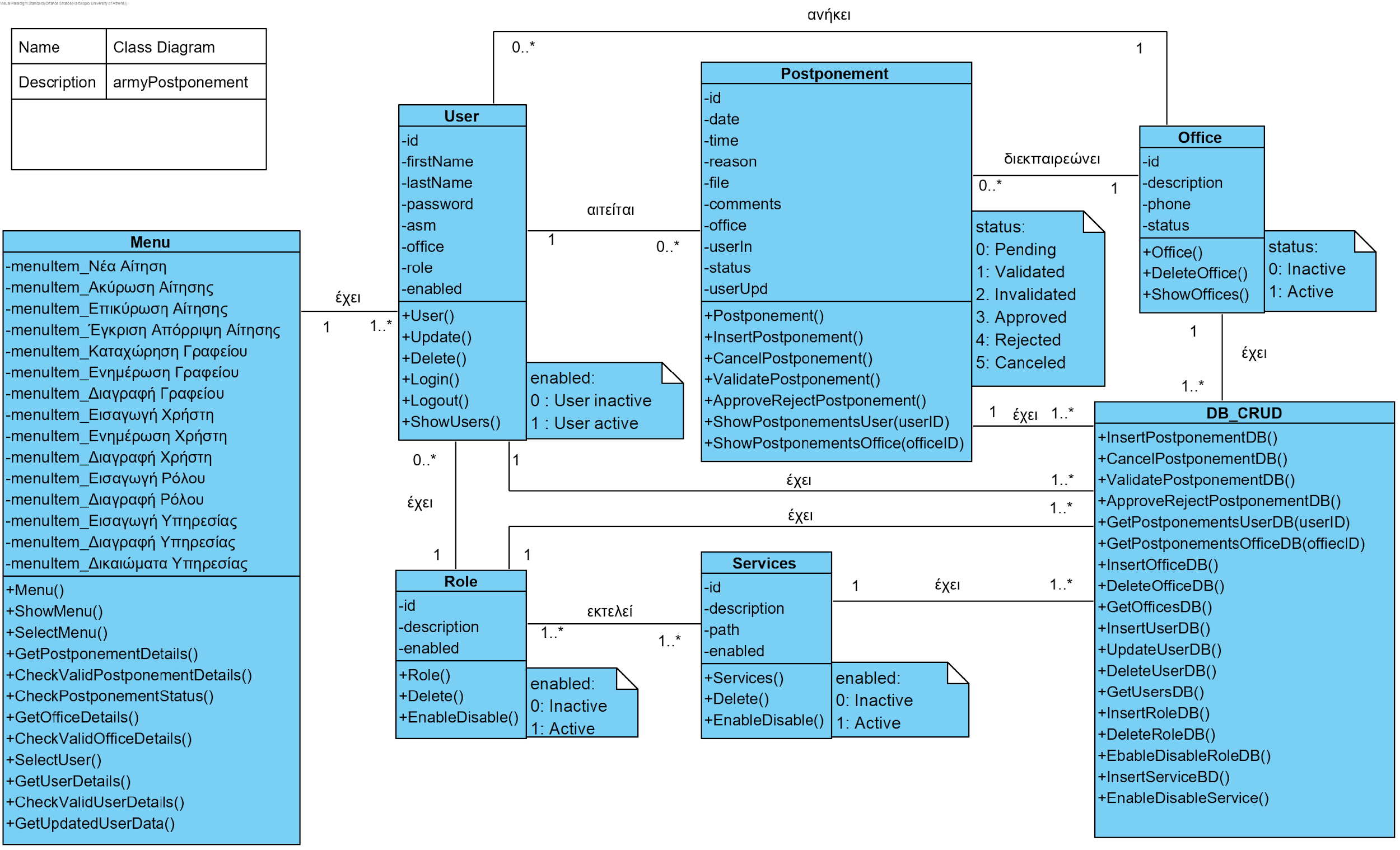
Στο παραπάνω διάγραμμα φαίνονται οι δύο λειτουργίες που αντιστοιχούν στη διαχείριση φορέων καθώς και οι δύο σχετικοί actors.

### 2.5.4 Use Case για την Διαχείριση καταλόγου χρηστών



Στο παραπάνω διάγραμμα φαίνονται οι οχτώ λειτουργίες που αντιστοιχούν στη διαχείριση χρηστών και διαχείριση ρόλων καθώς και οι δύο σχετικοί actors. Ο διαχειριστής συστήματος μπορεί να πραγματοποιήσει όλες τις λειτουργίες του διαγράμματος ενώ ο πολίτης μόνο μία από αυτές.

## 2.6 Class διάγραμμα



Στο παραπάνω class diagram απεικονίζονται όλες οι κλάσεις του συστήματος. Σε κάθε κλάση φαίνονται τα χαρακτηριστικά της (attributes) και οι μέθοδοι της (methods).

Αρχικά φαίνεται η κλάση Menu με όλα τα menu items και τις διαθέσιμες λειτουργίες της που έχουν να κάνουν με την εμφάνιση του μενού στο χρήστη, την εμφάνιση των οθονών για τις διάφορες λειτουργίες της εφαρμογής, ρουτίνες για την εισαγωγή στοιχείων, καθώς και ρουτίνες για το validation των στοιχείων που εισάγει ο χρήστης.

Η κλάση User περιέχει πληροφορίες για τους χρήστες του συστήματος καθώς και διάφορες λειτουργίες που μπορεί να κάνει ο κάθε χρήστης. Αξίζει να σημειωθεί το χαρακτηριστικό rοle που περιέχει τον κωδικό του ρόλου που έχει ο χρήστης στο σύστημα ο οποίος περιγράφεται στην κλάση role. Ο κάθε χρήστης μπορεί να έχει ένα ρόλο στο σύστημα. Το πεδίο enabled δείχνει εάν ένας υπάρχων χρήστης μπορεί να κάνει Login στην εφαρμογή. By default είναι 1. Εάν θέλουμε να σβήσουμε ένα χρήστη ή να τον κάνουμε disable τότε ο διαχειριστής καλώντας τη μέθοδο Delete αλλάζει το πεδίο σε 0. Γενικά δεν σβήνουμε χρήστες από το σύστημα, δεδομένου ότι μπορεί να έχουν ραντεβού που πρέπει να έχουν κάποιον owner. Απλά αλλάζουμε την τιμή του πεδίου enable σε 0 και έτσι δεν μπορούν να κάνουν το Login στο σύστημα.

Η κλάση Role περιέχει τους ρόλους του συστήματος. Τα χαρακτηριστικά είναι κωδικός, περιγραφή, και enabled που δείχνει αν είναι ενεργή ή όχι. Η μέθοδος Role() δημιουργεί ένα νέο ρόλο, η Delete() διαγράφει ένα ρόλο, ενώ η EnableDisable() ενεργοποιεί ή απενεργοποιεί ένα ρόλο.

Η κλαση Services περιέχει τις διαθέσιμες λειτουργίες του συστήματος προς τους χρήστες. Τα χαρακτηριστικά της είναι: κωδικός, περιγραφή και enabled (αν είναι ενεργή ή όχι). Οι μέθοδοι είναι: δημιουργία νέας υπηρεσίας, διαγραφή υπηρεσίας και EnableDisable υπηρεσίας. Μία υπηρεσία μπορεί να εκτελεστεί από πολλούς ρόλους όπως επίσης και ένας ρόλος μπορεί να εκτελεί πολλές υπηρεσίες. Αυτή η σχέση που είναι πολλά προς πολλά θα αποτυπωθεί στο ER diagram με ξεχωριστό πίνακα.

Η κλαση Appointment περιέχει πληροφορίες για τα ραντεβού. Τα χαρακτηριστικά είναι: κωδικός, ημερομηνία, ώρα, φορέας που έχουμε ραντεβού, σχόλια, χρήστης (Πολίτης) που αιτείται το ραντεβού, status του ραντεβού όπως εξηγείται στο παράρτημα της κλάσης στο class diagram και χρήστης που άλλαξε τελευταίος την κατάσταση του ραντεβού. Ένα ραντεβού όταν γίνεται εισαγωγή στο σύστημα το Status του είναι 0. Ο υπάλληλος του φορέα μπορεί να εγκρίνει (approve) ή να απορρίψει (reject) το ραντεβού επομένως το πεδίο status γίνεται 1 ή 2 αντίστοιχα. Ο πολίτης που έχει καταχωρήσει το ραντεβού μπορεί να ακυρώσει το ραντεβού εφόσον είναι σε κατάσταση νέο ή εγκεκριμένο και η ημερομηνία δεν έχει παρέλθει της τρέχουσας ώρας της ακύρωσης.

Η κλάση Foreas περιέχει πληροφορίες για τους φορείς στους οποίους μπορούν οι πολίτες να κλείσουν ραντεβού. Τα χαρακτηριστικά είναι: κωδικός, όνομα, υπεύθυνος φορέα ο οποίος πρέπει να είναι υπάρχον χρήστης του συστήματος και status πού παίρνει τιμές 0 για νέο φορέα, 1 για εγκεκριμένο φορέα και 2 για φορέα που έχει απορριφθεί. Την έγκριση ή την απόρριψη του φορέα την εκτελεί ο διαχειριστής φορέων. Είναι προφανές ότι οι πολίτες μπορούν να κλείσουν ραντεβού μόνο σε φορέα ο οποίος είναι εγκεκριμένος.

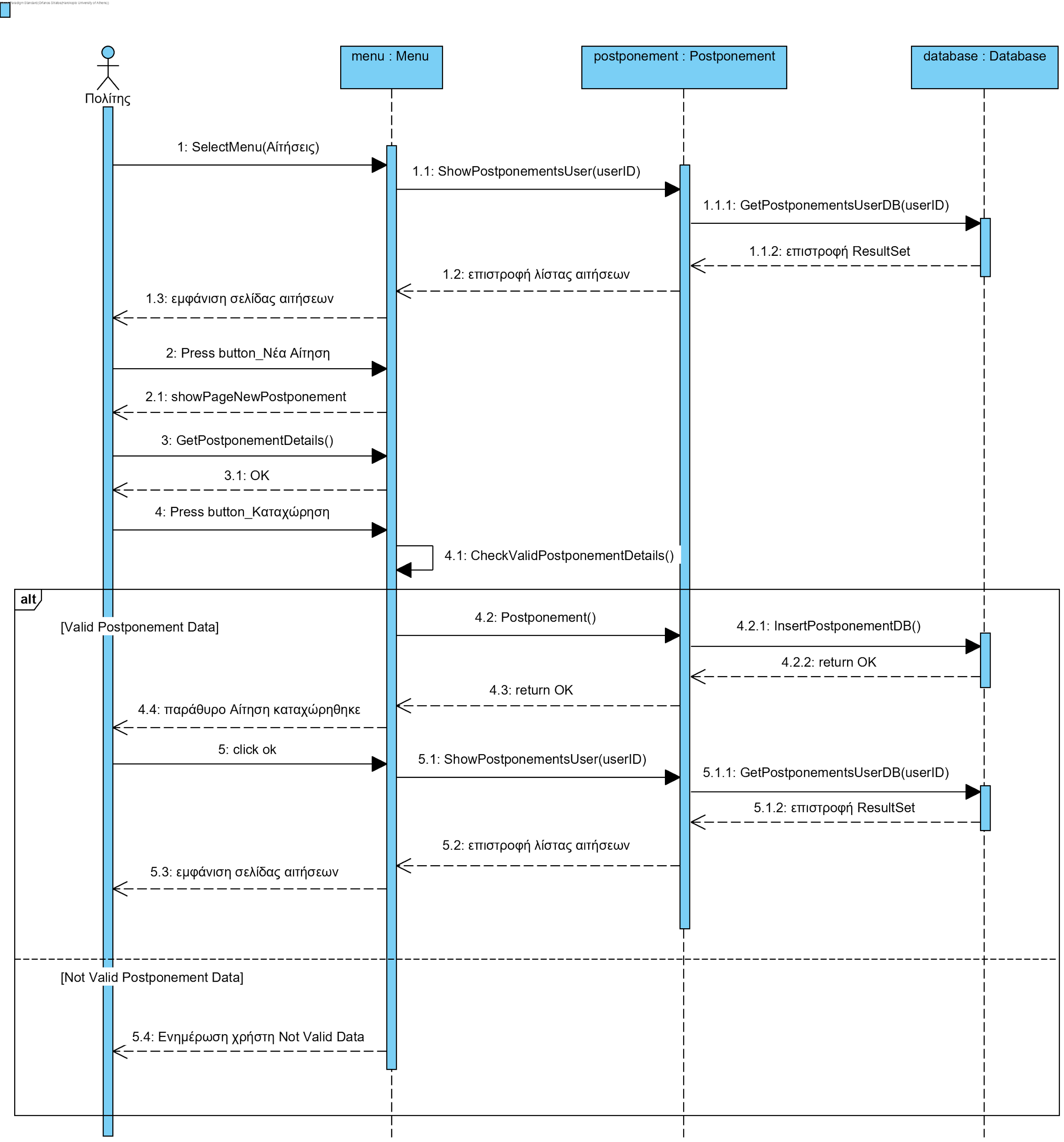
Τέλος υπάρχει η κλάση DB\_CRUD. Η κλάση αυτή δεν έχει χαρακτηριστικά αλλά μόνο static μεθόδους, τις οποίες οι υπόλοιπες κλάσεις του συστήματος καλούν κάθε φορά που θέλουν να επικοινωνήσουν με τη βάση για εισαγωγή νέων δεδομένων, ενημέρωση δεδομένων ή ανάκτηση δεδομένων που αφορούν την κλάση τους.

## 2.7 Sequence διαγράμματα

Παρακάτω παραθέτουμε τα sequence διαγράμματα των βασικών λειτουργιών της εφαρμογής.

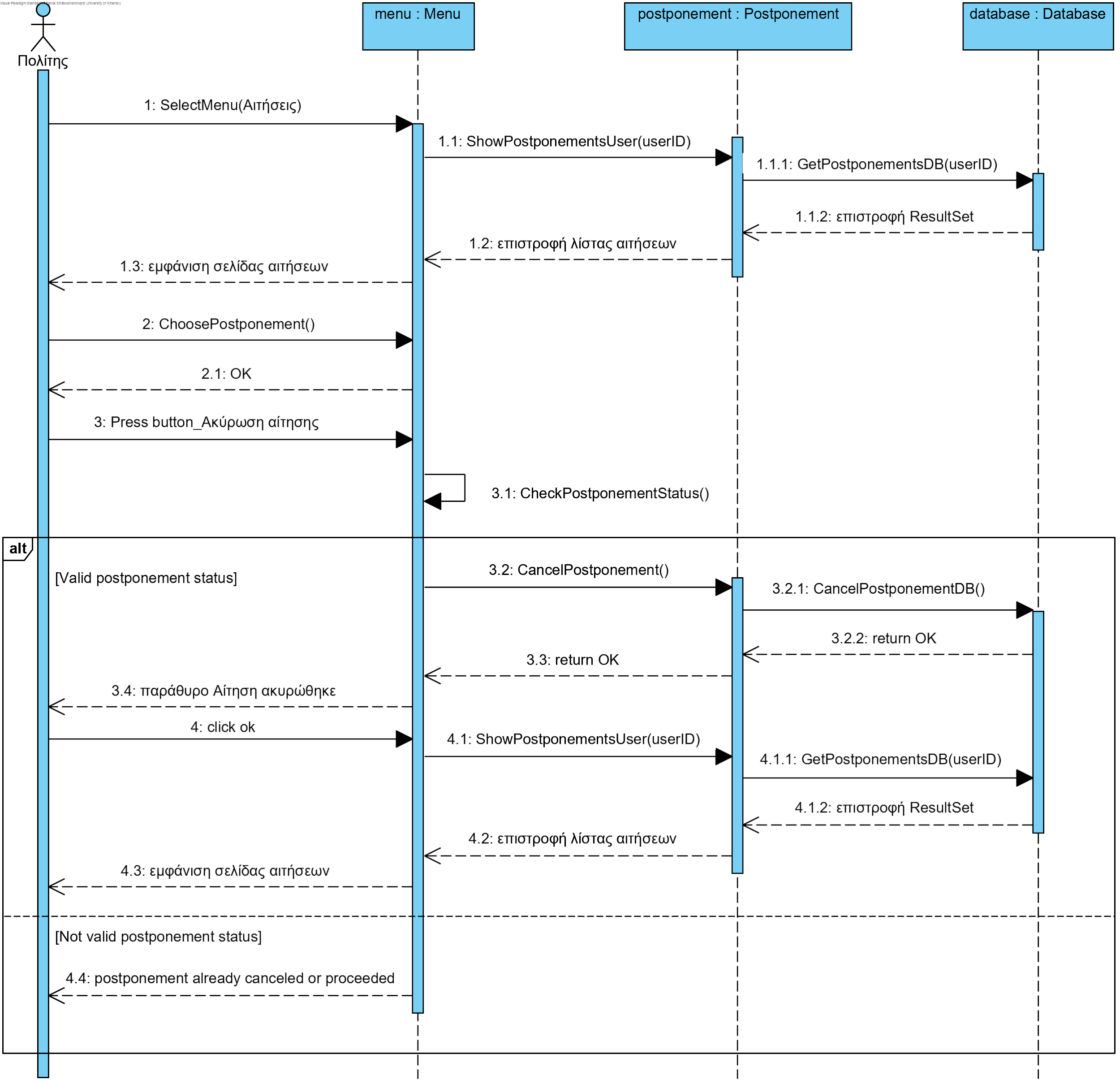
### 2.7.1 Πολίτης

* **Νέα αίτηση αναβολής στράτευσης**



Στο παραπάνω διάγραμμα παρουσιάζουμε τη δημιουργία νέου ραντεβού από τον πολίτη. Ο πολίτης διαλέγει «Νέο Ραντεβού» από το μενού επιλογών, το σύστημα επικοινωνεί με τη βάση δεδομένων και του εμφανίζει σε πρώτη φάση όλα τα ραντεβού του. Ο πολίτης, στη συνέχεια, πατάει «νέο ραντεβού» και του εμφανίζεται μια φόρμα στην οποία συμπληρώνει τα στοιχεία του ραντεβού. Τέλος, πατάει «καταχώρηση», το σύστημα ελέγχει τα στοιχεία και αν είναι έγκυρα, καταχωρείται στη βάση δεδομένων, αλλιώς τον ενημερώνει ότι έχει γίνει λάθος στα δεδομένα.

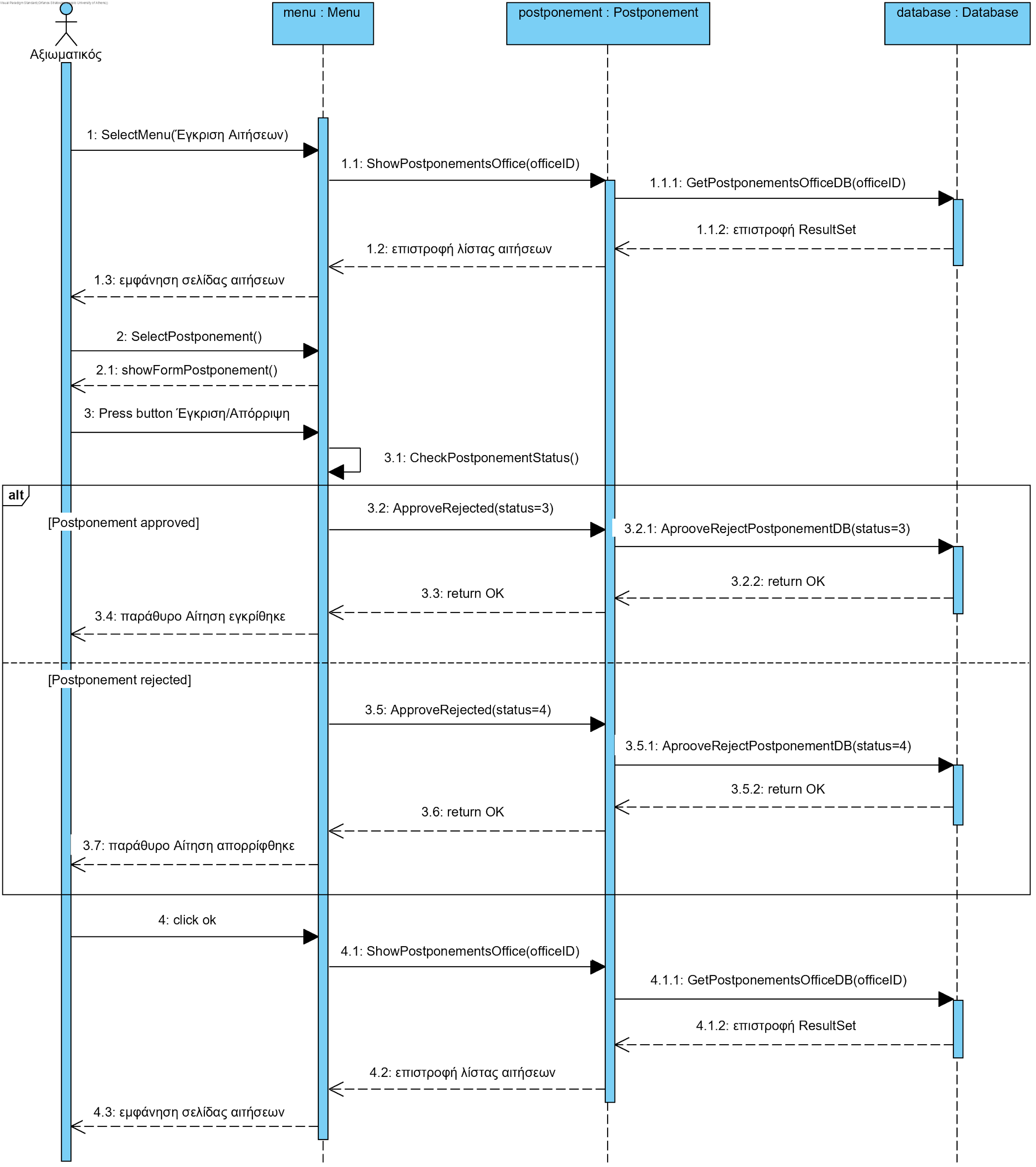
* **Ακύρωση αίτησης**



Στο παραπάνω διάγραμμα παρουσιάζουμε την ακύρωση ραντεβού από τον πολίτη. Ο πολίτης διαλέγει «Ακύρωση Ραντεβού» από το μενού επιλογών, το σύστημα επικοινωνεί με τη βάση δεδομένων και του εμφανίζει σε πρώτη φάση όλα τα ραντεβού του. Ο πολίτης, στη συνέχεια, επιλέγει το ραντεβού που θέλει να ακυρώσει και πατάει «Ακύρωση». Τέλος, εφόσον το ραντεβού που επιλέχθηκε δεν έχει ήδη ακυρωθεί ή πραγματοποιηθεί, το σύστημα επικοινωνεί με την βάση δεδομένων και αλλάζει το status του.

### 2.7.2 Αξιωματικός

* **Έγκριση/Απόρριψη αίτησης**



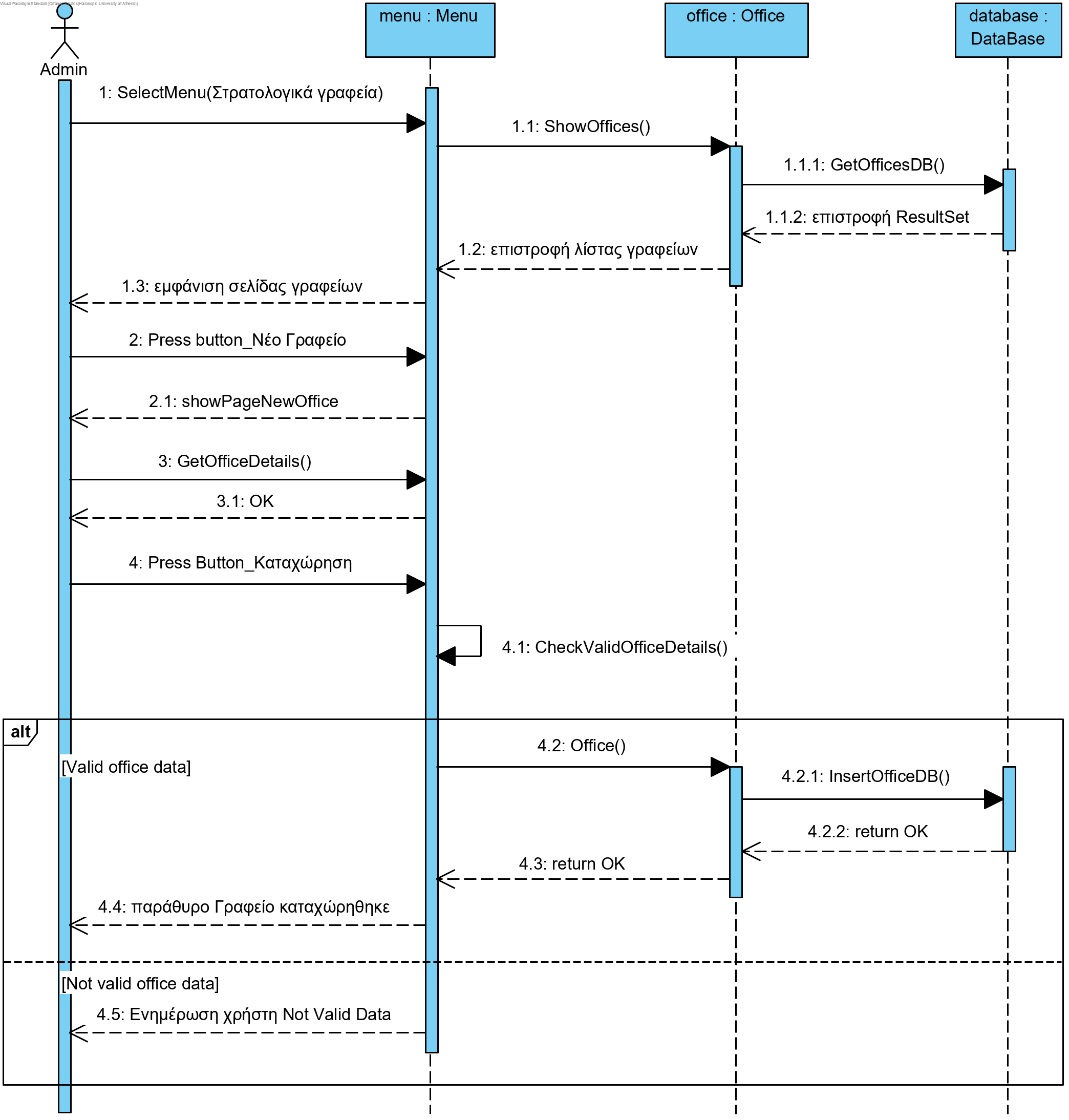
### 2.7.3 Υπάλληλος

* **Επικύρωση/Μη επικύρωση αίτησης**

ιδιο με το παραπάνω

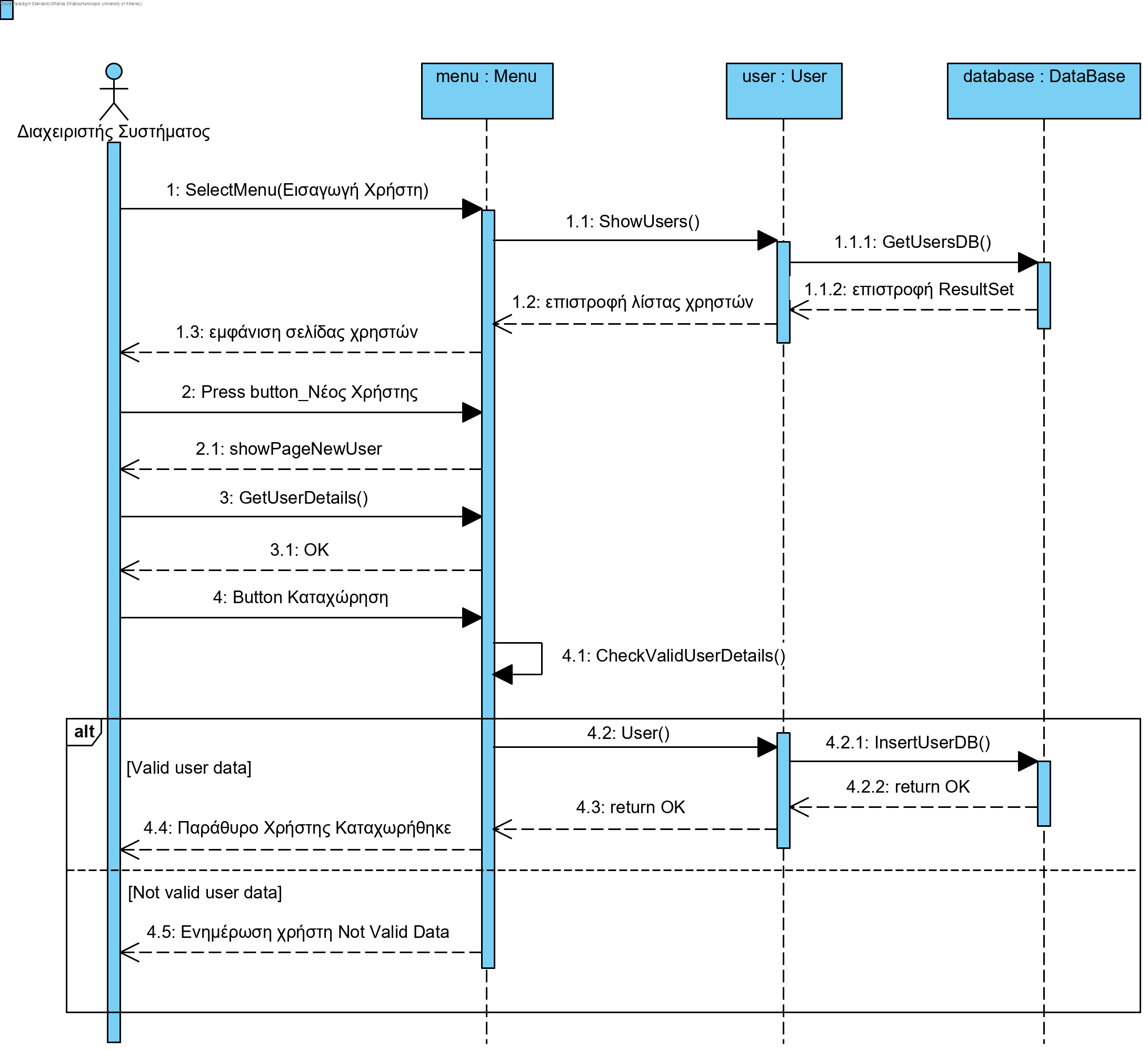
### 2.7.4 Διαχειριστής Συστήματος

* **Καταχώρηση στρατολογικού γραφείου**

Στο παραπάνω διάγραμμα ο διαχειριστής συστήματος επιλέγει στο μενού «εισαγωγή ρόλου», όπου θα του εμφανιστεί σχετική φόρμα και θα συμπληρώσει τις πληροφορίες που απαιτούνται. Θα πατήσει το κουμπί για καταχώρηση και αν οι πληροφορίες είναι έγκυρες θα καταχωρηθεί στην βάση και θα ενημερωθεί με σχετικό μήνυμα, αλλιώς θα λάβει ειδοποίηση ότι τα δεδομένα ήταν λανθασμένα.

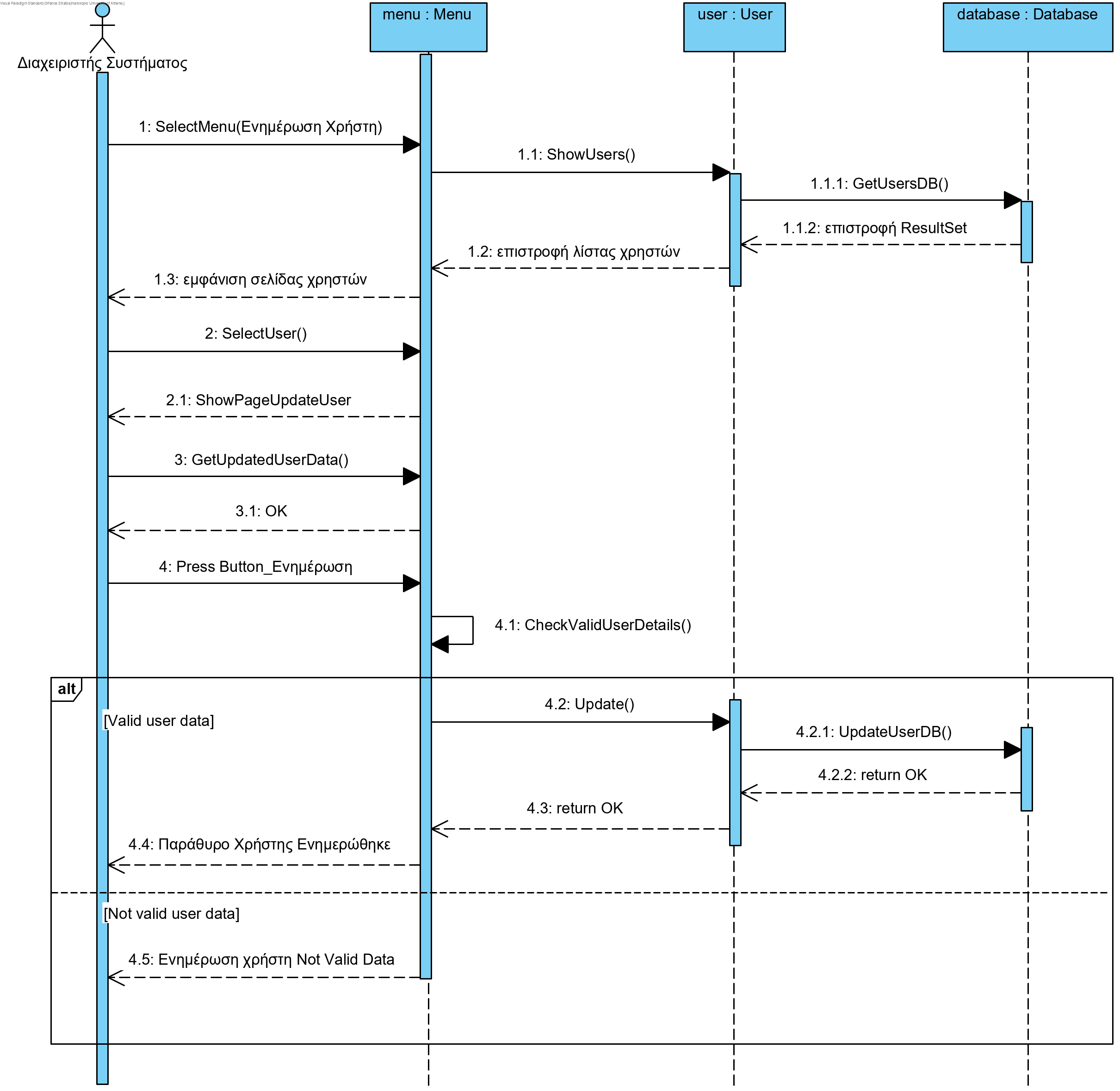
Στο παραπάνω διάγραμμα παρουσιάζουμε το αίτημα καταχώρησης ενός φορέα από τον προϊστάμενο φορέα στο σύστημα. Ο προϊστάμενος εισέρχεται στο σύστημα επιλέγει «αίτημα καταχώρησης φορέα» και του εμφανίζεται μια φόρμα όπου καταχωρεί τα στοιχεία του φορέα. Πατάει το button καταχώρηση και εκεί το σύστημα ελέγχει αν είναι έγκυρα τα στοιχεία. Αν είναι έγκυρα το αίτημα καταχωρείται στην βάση δεδομένων, εναλλακτικά εμφανίζεται σχετική ειδοποίηση.

* **Εισαγωγή χρήστη**



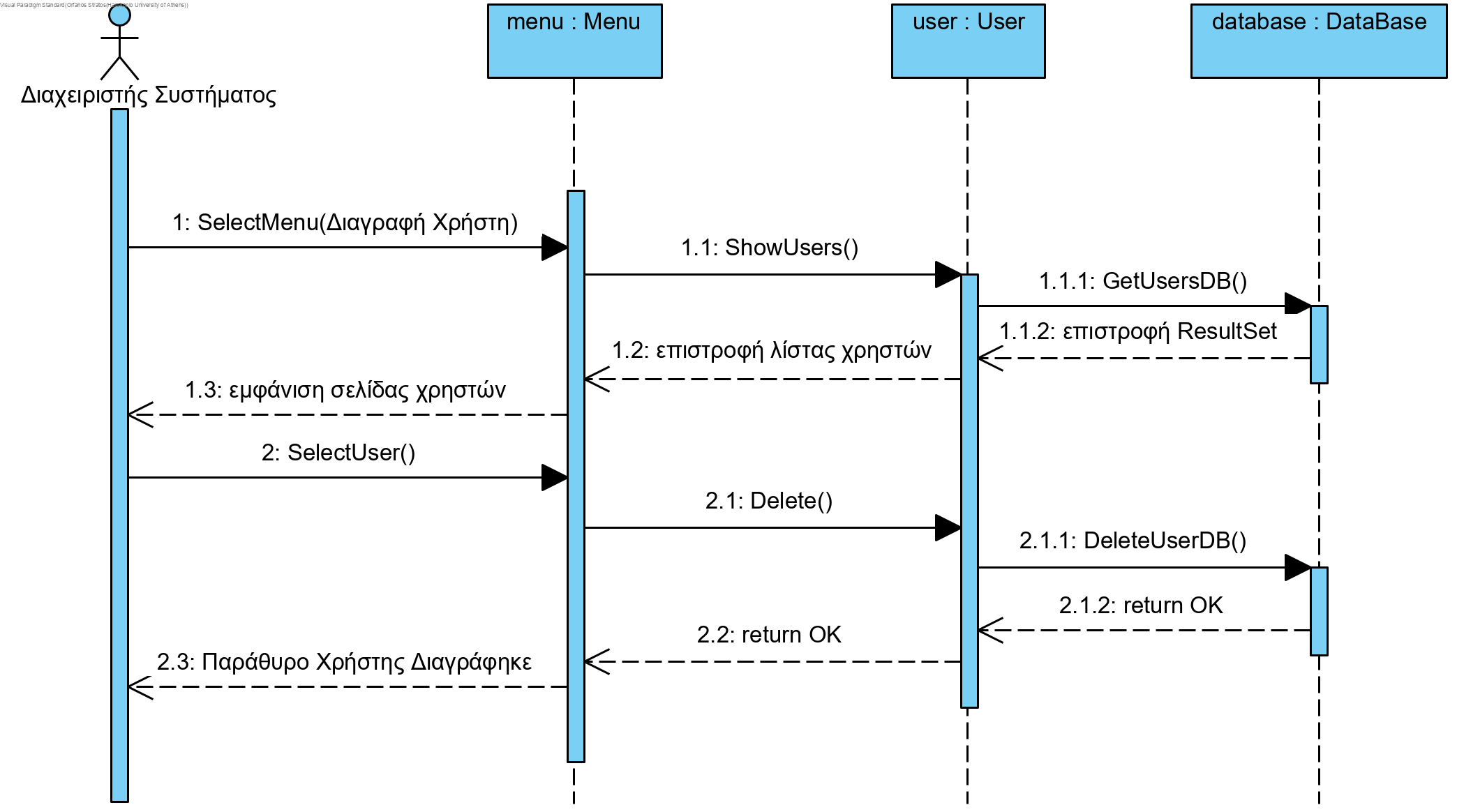
Στο παραπάνω διάγραμμα παρουσιάζεται η εισαγωγή νέου χρήστη από τον διαχειριστή. Ο διαχειριστής επιλέγει «Εισαγωγή Χρήστη», το σύστημα επικοινωνεί με την βάση δεδομένων και επιστρέφει μια λίστα με όλους τους χρήστες. Στη συνέχεια, πατάει «Νέος Χρήστης» όπου του εμφανίζεται μια φόρμα στην οποία συμπληρώνει τα στοιχεία του χρήστη και πατάει καταχώριση. Τέλος, το σύστημα ελέγχει τα στοιχεία και, αν είναι έγκυρα, ο χρήστης αποθηκεύεται στη βάση δεδομένων, αλλιώς ενημερώνεται με σχετική ειδοποίηση.

* **Ενημέρωση χρήστη**



Στο παραπάνω διάγραμμα παρουσιάζεται η ενημέρωση ενός χρήστη από τον διαχειριστή του συστήματος. Ο διαχειριστής διαλέγει «Ενημέρωση Χρήστη» από το μενού επιλογών και το σύστημα του εμφανίζει τους χρήστες του συστήματος μετά από επικοινωνία με τη βάση δεδομένων. Στη συνέχεια, ο διαχειριστής επιλέγει τον χρήστη που επιθυμεί. Το σύστημα του εμφανίζει μια φόρμα με τα στοιχεία του χρήστη και όταν ο διαχειριστής πραγματοποιήσει τις αλλαγές πατάει «Ενημέρωση». Τέλος, ελέγχεται εάν τα στοιχεία είναι έγκυρα και εφόσον είναι, ενημερώνεται η βάση δεδομένων, αλλιώς επιστρέφεται αντίστοιχη ειδοποίηση.

* **Διαγραφή χρήστη**

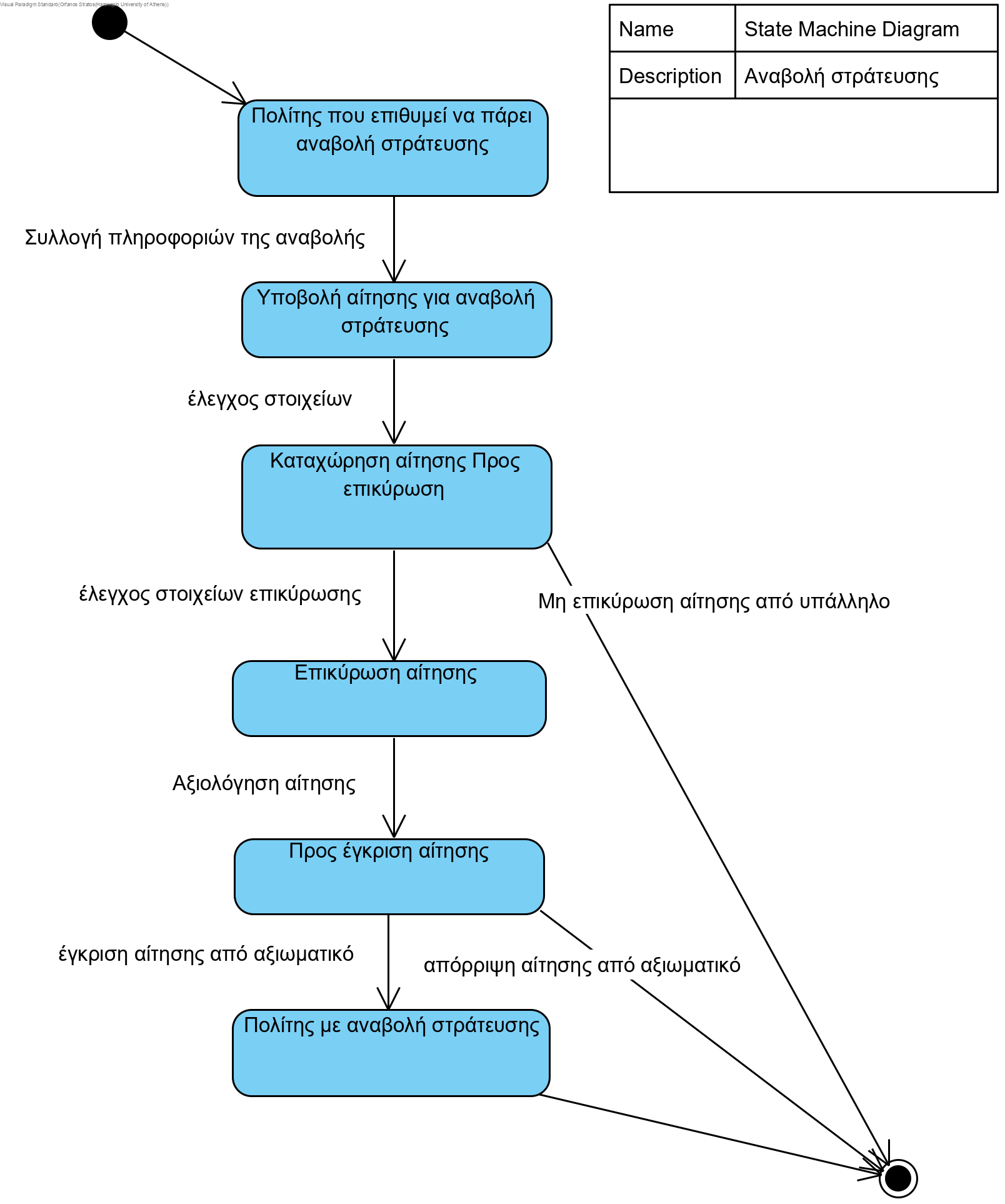


Στο παραπάνω διάγραμμα ο διαχειριστής συστήματος, επιλέγει στο μενού «διαγραφή χρήστη», όπου εκεί από την βάση δεδομένων του εμφανίζεται μια λίστα χρηστών. Θα επιλέξει τον χρήστη που επιθυμεί και αυτός θα διαγραφεί, γεγονός που θα αποτυπωθεί και στην βάση δεδομένων. Τέλος θα ενημερωθεί ότι η διαδικασία ήταν επιτυχής και σε σχετικό μήνυμα.

Στο παραπάνω διάγραμμα ο διαχειριστής συστήματος επιλέγει στο μενού «εισαγωγή ρόλου», όπου θα του εμφανιστεί σχετική φόρμα και θα συμπληρώσει τις πληροφορίες που απαιτούνται. Θα πατήσει το κουμπί για καταχώρηση και αν οι πληροφορίες είναι έγκυρες θα καταχωρηθεί στην βάση και θα ενημερωθεί με σχετικό μήνυμα, αλλιώς θα λάβει ειδοποίηση ότι τα δεδομένα ήταν λανθασμένα.

## 2.8 State machine διαγράμματα

2.8.1 State machine διαγράμμα: Αναβολή στράτευσης



Στο παραπάνω διάγραμμα επιλέξαμε να αποτυπώσουμε την βασική λειτουργεία του συστήματος που είναι η καταχώρηση ραντεβού από έναν χρήστη του συστήματος. Έχουμε λοιπόν έναν χρήστη που επιθυμεί να κλείσει ραντεβού, θα δηλώσει τις απαραίτητες πληροφορίες και από εκεί περνάμε στην κατάσταση όπου θα γίνει η αίτηση. Τα στοιχεία θα επεξεργασθούν και θα τεθούν υπό έλεγχο και αφού γίνουν οι απαραίτητοι έλεγχοι το ραντεβού εν τέλει θα εγκριθεί ή θα απορριφθεί.

## 2.9 Component διάγραμμα

Το διάγραμμα αυτό θα παραδοθεί σε επόμενο παραδοτέο.

## 2.10 Deployment διάγραμμα

Το διάγραμμα αυτό θα παραδοθεί σε επόμενο παραδοτέο.

# Τεύχος υλοποίησης της εφαρμογής

# Εγχειρίδιο χρήσης