



30 ΣΕΠΤΕΜΒΡΙΟΥ 2022

# ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ


ΕΙΔΙΚΟ ΘΕΜΑ ΟΜΑΔΑΣ 3

Συντελεστές εργασίας

Χριστοφορίδης Χαράλαμπος – Π19188

Μίσσας Ιωάννης – Π19105

Καρκάνης Ευστράτιος - Π19064



## Πίνακας περιεχομένων

1. Εισαγωγή .....	3
1.1 Στόχοι της εργασίας .....	3
1.2 Ορισμός του προβλήματος προς επίλυση .....	3
2. Σύντομη παρουσίαση της RUP (Rational Unified Process).....	3
3. Φάση: Έναρξη (Inception) .....	4
3.1 Σύλληψη απαιτήσεων .....	4
3.2 Ανάλυση-Σχεδιασμός .....	5
3.2.1 Διαγράμματα Περιπτώσεων Χρήσης .....	5
3.2.2 Διαγράμματα Τάξεων.....	5
4. Φάση: Εκπόνηση Μελέτης (Elaboration).....	6
4.1 Ανάλυση-Σχεδιασμός .....	6
4.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2η έκδοση).....	6
4.1.2 Διαγράμματα Τάξεων (2η έκδοση) .....	7
4.1.3 Διαγράμματα Αντικειμένων (1η έκδοση) .....	8
4.1.4 Διαγράμματα Συνεργασίας (1η έκδοση) .....	8
4.1.5 Διαγράμματα Σειράς (1η έκδοση) .....	9
4.1.6 Διαγράμματα Δραστηριοτήτων (1η έκδοση).....	10
4.1.7 Διαγράμματα Καταστάσεων (1η έκδοση).....	11
4.1.8 Διαγράμματα Εξαρτημάτων (1η έκδοση) .....	12
4.1.9 Διαγράμματα Διανομής (1η έκδοση).....	13
4.2 Υλοποίηση-Έλεγχος .....	14
4.2.1 Υλοποίηση: 1η εκτελέσιμη έκδοση.....	14
4.2.2 Αναφορά ελέγχου για την 1η εκτελέσιμη έκδοση.....	15
5. Φάση: Κατασκευή (Construction).....	15
5.1 Ανάλυση-Σχεδιασμός .....	15
5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3η έκδοση).....	15
5.1.2 Διαγράμματα Τάξεων (3η έκδοση) .....	16
5.1.3 Διαγράμματα Αντικειμένων (2η έκδοση) .....	17
5.1.4 Διαγράμματα Συνεργασίας (2η έκδοση) .....	17
5.1.5 Διαγράμματα Σειράς (2η έκδοση) .....	18
5.1.6 Διαγράμματα Δραστηριοτήτων (2η έκδοση).....	19
5.1.7 Διαγράμματα Καταστάσεων (2η έκδοση).....	20
5.1.8 Διαγράμματα Εξαρτημάτων (2η έκδοση) .....	21
5.1.9 Διαγράμματα Διανομής (2η έκδοση).....	22

5.2	Υλοποίηση-Έλεγχος .....	22
5.2.1	Υλοποίηση: Τελική εκτελέσιμη έκδοση .....	22
5.2.2	Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση .....	24
6.	Εγχειρίδιο Χρήστη .....	25
6.1	Σύντομη παρουσίαση του προγράμματος .....	25
6.2	Παρουσίαση σεναρίων λειτουργίας .....	30
7.	Βιβλιογραφικές Πηγές .....	35

## 1. Εισαγωγή

### 1.1 Στόχοι της εργασίας

Οι στόχοι της εργασίας είναι να φτιάξουμε μια εφαρμογή τύπου ημερολογίου/ατζέντας, στην οποία θα μπορεί ο χρήστης να συνδεθεί για να κάνει τις ακόλουθες λειτουργίες :

- Σύνδεση και αποσύνδεση του χρήστη από το σύστημα.
- Εισαγωγή και διαγραφή ενός ραντεβού στην πλατφόρμα από τον συνδεδεμένο χρήστη.
- Τροποποίηση υπάρχοντων ραντεβού που έχει φτιάξει ο συνδεδεμένος χρήστης.
- Προβολή όλων των ραντεβού που έχει ο συνδεδεμένος χρήστης.
- Προσθήκη και άλλων ατόμων σε ένα ραντεβού που έχει δημιουργήσει ο συνδεδεμένος χρήστης.

### 1.2 Ορισμός του προβλήματος προς επίλυση

Το πρόβλημα είναι να φτιάξουμε ένα λογισμικό που θα βοηθάει τον χρήστη να οργανώσει την καθημερινότητά του, δηλαδή ένα διαδικτυακό ημερολόγιο. Στην ουσία αποτελεί ένα λογισμικό τόσο λειτουργικό, όσο και εύχρηστο. Η εφαρμογή επιτρέπει στον χρήστη να έχει πρόσβαση ανά πάσα στιγμή στο πρόγραμμα της ημέρας του, καθώς και να κάνει τις απαραίτητες αλλαγές σε αυτό όπως εκείνος επιθυμεί.

## 2. Σύντομη παρουσίαση της RUP (Rational Unified Process)

Η RUP είναι μια διαδικασία τεχνολογίας λογισμικού (Software Engineering Process). Ο στόχος της είναι να διασφαλίσει την παραγωγή λογισμικού υψηλής ποιότητας που ικανοποιεί τις ανάγκες των τελικών χρηστών μέσα σε ένα συγκεκριμένο χρονοδιάγραμμα και κόστος.

Ο κύκλος ζωής του λογισμικού (Software life-cycle) σκιαγραφεί τη ζωή του προγράμματος λογισμικού από τη στιγμή της γέννησής του μέχρι τη στιγμή της αντικατάστασης ή της εγκατάλειψής του.

Ο κύκλος ζωής του λογισμικού στην RUP υποδιαιρείται σε τέσσερις συνεχόμενες φάσεις. Οι τέσσερις φάσεις είναι:

- η φάση σύλληψης (Inception Phase),
- η φάση επεξεργασίας (Elaboration Phase),
- η φάση κατασκευής (Construction Phase) και
- η φάση μετάβασης (Transition Phase)

Ορισμοί σχετικοί με την RUP:

- **Κύκλος ανάπτυξης:** Ένα πέρασμα από τις τέσσερις φάσεις. Κάθε τέτοιο πέρασμα

παράγει μια νέα γενιά (generation) λογισμικού.

- **Κύκλος εξέλιξης:** Οι διαδοχικοί κύκλοι στην περίπτωση που το προϊόν συνεχίσει να εξελίσσεται στην επόμενη γενιά του, ονομάζονται κύκλοι εξέλιξης (evolution cycles).

### 3. Φάση: Έναρξη (Inception)

#### 3.1 Σύλληψη απαιτήσεων

Παρακάτω καταγράφονται οι λειτουργικές απαιτήσεις της εφαρμογής, δηλαδή οι υπηρεσίες που πρέπει να παρέχει το σύστημα και πως εκείνο πρέπει να αντιδρά στις διάφορες καταστάσεις και εισόδους:

1. να προσαρμόζεται η εφαρμογή στο μέγεθος της οθόνης του χρήστη, είτε είναι υπολογιστής, είτε είναι smartphone.
2. να αποθηκεύει τους χρήστες σε μία βάση δεδομένων (δυνατότητα αποθήκευσης στοιχείων).
3. να μπορεί να γίνει δημιουργία λογαριασμού από ένα νέο χρήστη.
4. να τακτοποιείται ένας χρήστης όταν υποβάλει τα στοιχεία του (username και κωδικός).
5. όλα τα ραντεβού θα πρέπει να αποθηκεύονται επίσης στην βάση δεδομένων.
6. δυνατότητα σύνδεσης και αποσύνδεσης στο σύστημα με εντολή του χρήστη.
7. να γίνεται εισαγωγή και διαγραφή ραντεβού από τον χρήστη. Η διαγραφή γίνεται μόνο από εκείνον τον χρήστη που δημιούργησε το ραντεβού.
8. να μπορεί να γίνει τροποποίηση των στοιχείων του ραντεβού (ώρα, ημερομηνία, συμμετέχοντες, διάρκεια) μόνο από τον χρήστη που έχει δημιουργήσει το συμβάν.
9. να μπορούν να προστεθούν και άλλα άτομα σε ένα ραντεβού από τον χρήστη που έφτιαξε το ραντεβού αυτό.
10. να ειδοποιεί η πλατφόρμα τους συμμετέχοντες χρήστες σε περίπτωση αλλαγής/τροποποίησης του ενός ραντεβού από τον δημιουργό του.
11. σε περίπτωση που ένας χρήστης δεν ταυτοποιείται ή υπάρχει πρόβλημα επικοινωνίας της εφαρμογής με τη βάση δεδομένων, το πρόγραμμα στέλνει αντίστοιχα μηνύματα στον χρήστη.

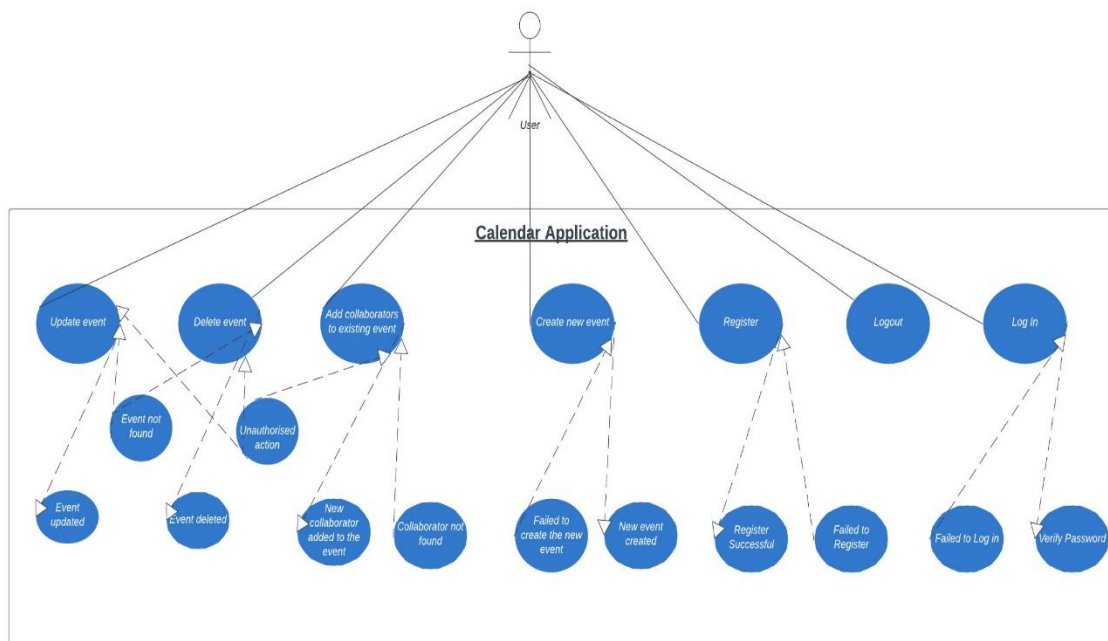
## 3.2 Ανάλυση-Σχεδιασμός

### 3.2.1 Διαγράμματα Περιπτώσεων Χρήσης

Σε αυτό το διάγραμμα περιγράφουμε τις λειτουργίες που είναι διαθέσιμες στον χρήστη όπως ορίζονται από την εκφώνηση, καθώς και οι μετέπειτα καταλήξεις αυτών. Συγκεκριμένα ο χρήστης μπορεί να:

1. Ενημερώσει ένα ραντεβού
2. Διαγράψει ένα ραντεβού
3. Προσθέσει συμμετέχοντες σε ένα ραντεβού
4. Δημιουργήσει ένα ραντεβού
5. Κάνει εγγραφή
6. Κάνει έξοδο από την εφαρμογή
7. Κάνει είσοδο στην εφαρμογή

Η φορά από τα βέλη δείχνει την χρονική συνέχεια εκτέλεσης. Για παράδειγμα ο χρήστης δεν μπορεί να κάνει Verify Password αν δεν πετύχει η λειτουργία Login. Παρ' όλα αυτά, το Failed to Login μπορεί να πραγματοποιηθεί ανεξάρτητα από την επιτυχία του Login.

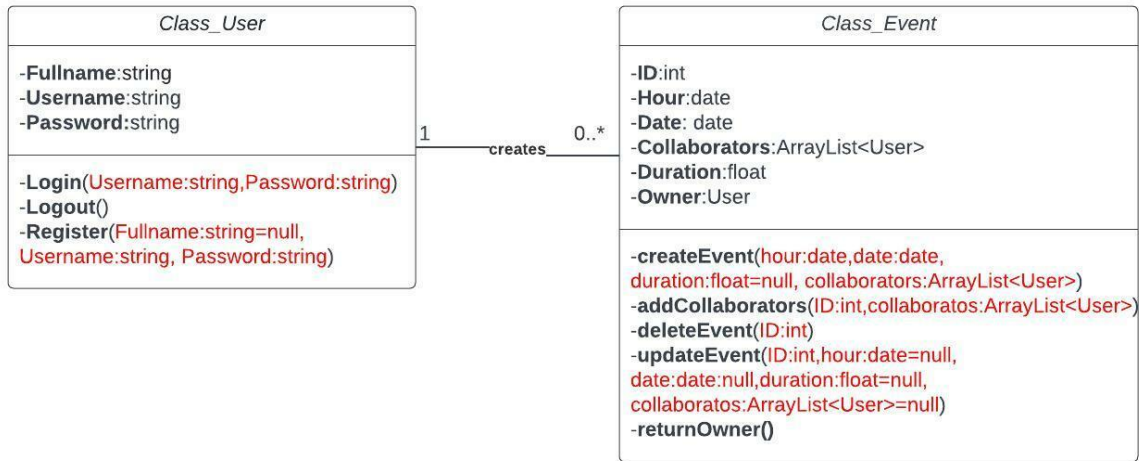


Διάγραμμα Περιπτώσεων Χρήσης – Έκδοση 1

### 3.2.2 Διαγράμματα Τάξεων

Σε αυτό το διάγραμμα αναλύουμε τις κλάσεις που θα χρησιμοποιήσουμε στην εφαρμογή μας. Έχουμε αποφασίσει να φτιάξουμε αρχικά δύο κλάσεις με τα παρακάτω

χαρακτηριστικά και συναρτήσεις. Οι κλάσεις αυτές συνδέονται καθώς τα αντικείμενα Event φτιάχνονται από τα αντικείμενα User. Ένα αντικείμενο User μπορεί να φτιάξει όσα Event επιθυμεί



Διάγραμμα Κλάσεων – Έκδοση 1

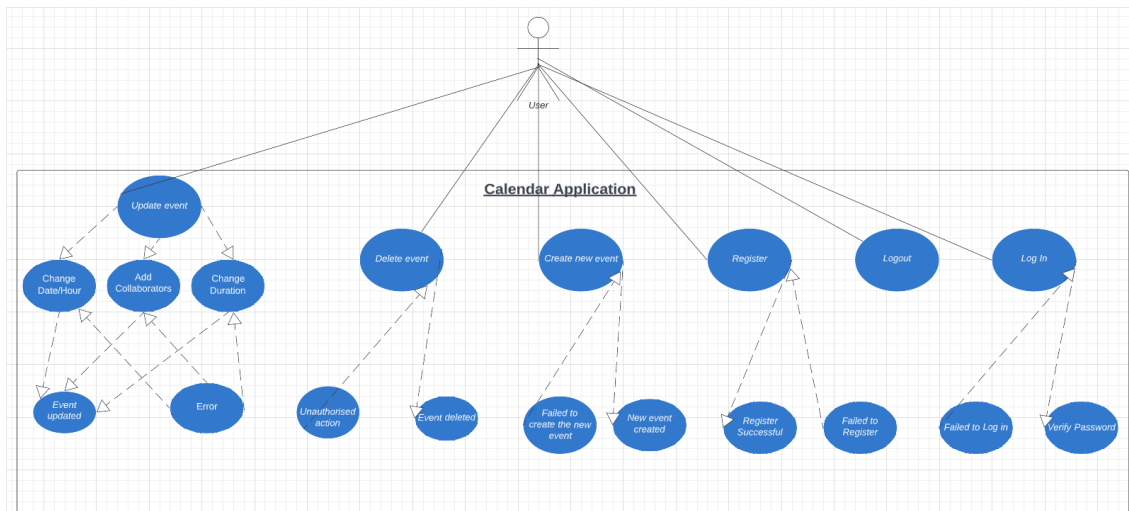
## 4. Φάση: Εκπόνηση Μελέτης (Elaboration)

### 4.1 Ανάλυση-Σχεδιασμός

#### 4.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2η έκδοση)

Στην 2η έκδοση του διαγράμματος περιπτώσεων χρήστη χρειάστηκε να:

- διαγράψουμε το Event not found μιας και η διαγραφή σαν λειτουργία εμφανίζεται μονάχα στα υπάρχοντα ραντεβού.
- διαγράψαμε το Unauthorised Action στο delete Event μια και δεν υπάρχει περίπτωση η διαγραφή να μην είναι έγκυρη, αφού εμφανίζεται μόνο στα ήδη υπάρχοντα ραντεβού.
- φτιάξαμε το update Event όπως ακολουθώντας τα πρότυπα της εφαρμογής, δηλαδή πρώτα διαλέγει ο χρήστης να κάνει Update και μετά κάνει αλλαγή σε συγκεκριμένο πεδίο του ραντεβού, άρα δεν είναι λειτουργίες που του δίνονται απευθείας.

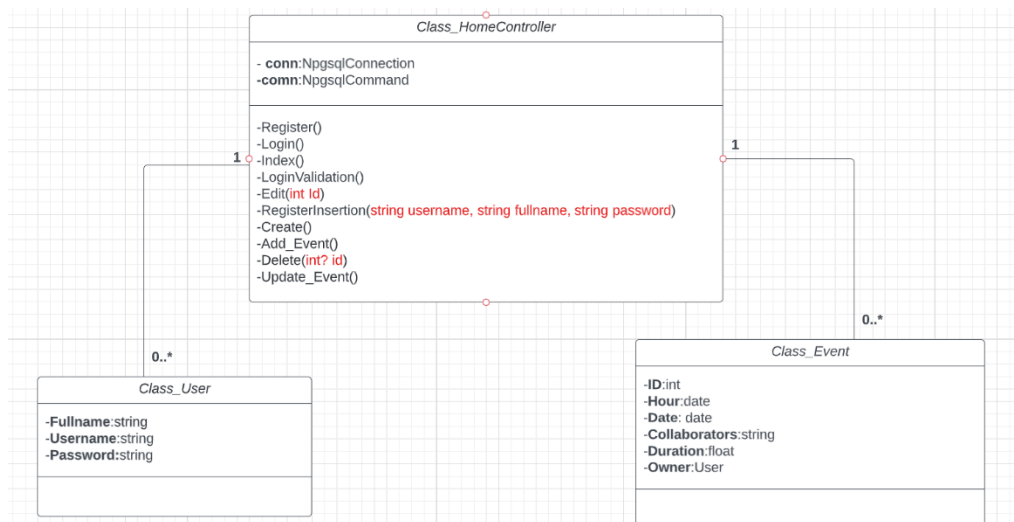


Διάγραμμα Περιπτώσεων Χρήσης - Έκδοση 2

#### 4.1.2 Διαγράμματα Τάξεων (2η έκδοση)

Στην 2η έκδοση του διαγράμματος τάξεων:

- χρειάστηκε να βγάλουμε τις συναρτήσεις από τις κλάσεις User και Event καθώς πλέον αυτές εκτελούνται από τον HomeController, τον οποίο και προσθέσαμε μαζί με τις νέες συναρτήσεις.
- αλλάξαμε τον τύπο του πεδίου Collaborators σε String (από String[]) καθώς ο προηγούμενος τύπος δεν μας εξυπηρετούσε στην εφαρμογή.

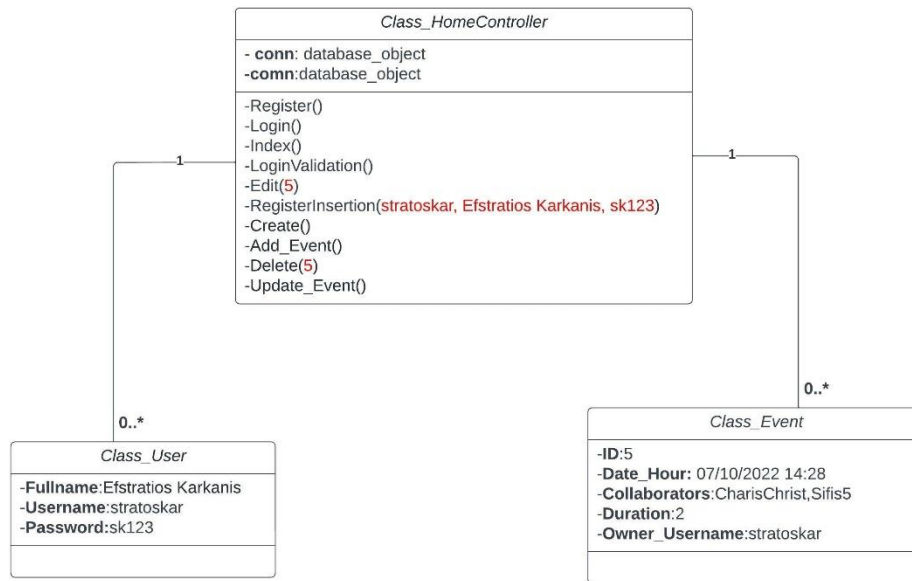


Διάγραμμα Τάξεων – Έκδοση 2



#### 4.1.3 Διαγράμματα Αντικειμένων (1η έκδοση)

Το διάγραμμα αντικειμένων περιγράφει μία δεδομένη κατάσταση του προβλήματος. Το διάγραμμα αυτό είναι πανομοιότυπο με το διάγραμμα τάξεων (απεικονίζονται σε αυτό τα αντικείμενα και οι τιμές που έχουν λάβει σε μία δεδομένη χρονική στιγμή). Για παράδειγμα, στο διάγραμμα παρακάτω, έχουν δημιουργηθεί τρία αντικείμενα User, HomeController και Event, με συγκεκριμένες τιμές στα attributes τους.

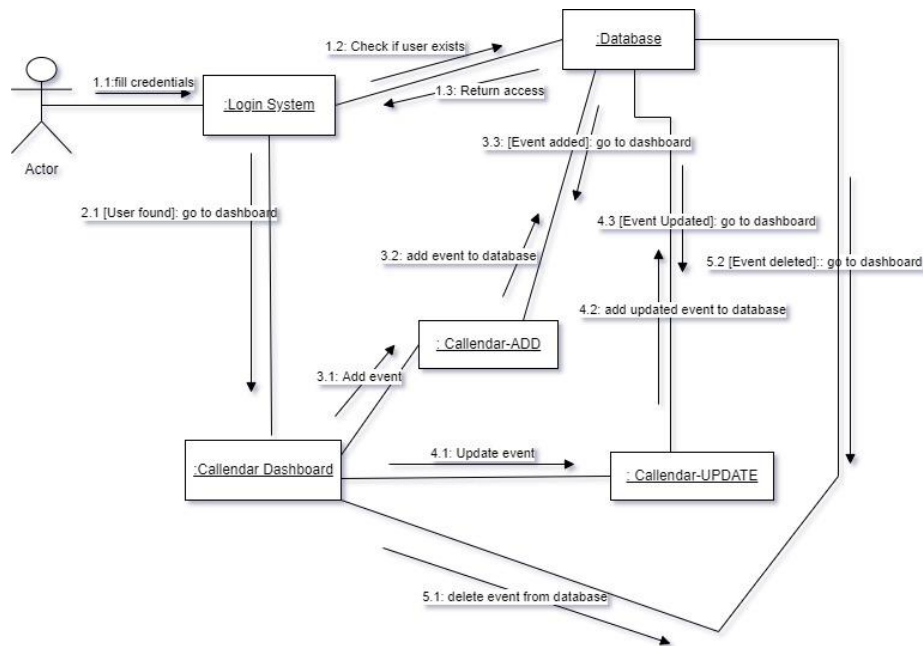


Διάγραμμα Αντικειμένων - Έκδοση 1

#### 4.1.4 Διαγράμματα Συνεργασίας (1η έκδοση)

Το διάγραμμα συνεργασίας δείχνει πως συνεργάζονται τα αντικείμενα της εφαρμογής μεταξύ τους. Προφανώς, για την συνεργασία αυτή, θα πρέπει να επέμβει και ο χρήστης. Για αυτόν τον λόγο απεικονίζεται στο διάγραμμα της εικόνας. Στο παρακάτω διάγραμμα, τα αντικείμενα (απεικονίζονται με ορθογώνια σχήματα) της εφαρμογής είναι το σύστημα login, η βάση δεδομένων, το μενού της εφαρμογής, η σελίδα για προσθήκη ενός ραντεβού και η σελίδα για ενημέρωση ενός ραντεβού. Οι γραμμές που ενώνουν τα αντικείμενα μεταξύ τους ονομάζονται σύνδεσμοι, ενώ πάνω από τους συνδέσμους βρίσκονται τα μηνύματα (αναπαρίστανται με βέλη που δείχνουν τη φορά αποστολής κάθε μηνύματος) που στέλνονται από ένα αντικείμενο σε ένα άλλο.

Ιδιαίτερη προσοχή φέρουν οι αριθμοί που βρίσκονται πριν από κάθε μήνυμα (1.1,1.2,1.3,...2.1,3.1,3.2...) καθώς αυτοί υποδηλώνουν την σειρά με την οποία στέλνονται τα μηνύματα. Στο εν λόγω διάγραμμα, πρώτα ο χρήστης εισάγει τα στοιχεία του (1.1), μετά το σύστημα login ελέγχει εάν ο χρήστης υπάρχει στέλνοντας μήνυμα στην βάση δεδομένων (1.2). Στην συνέχεια, εάν γίνει ταυτοποίηση του χρήστη, το σύστημα login αποστέλλει το μήνυμα υπ' αριθμόν 2.1, κ.ο.κ.

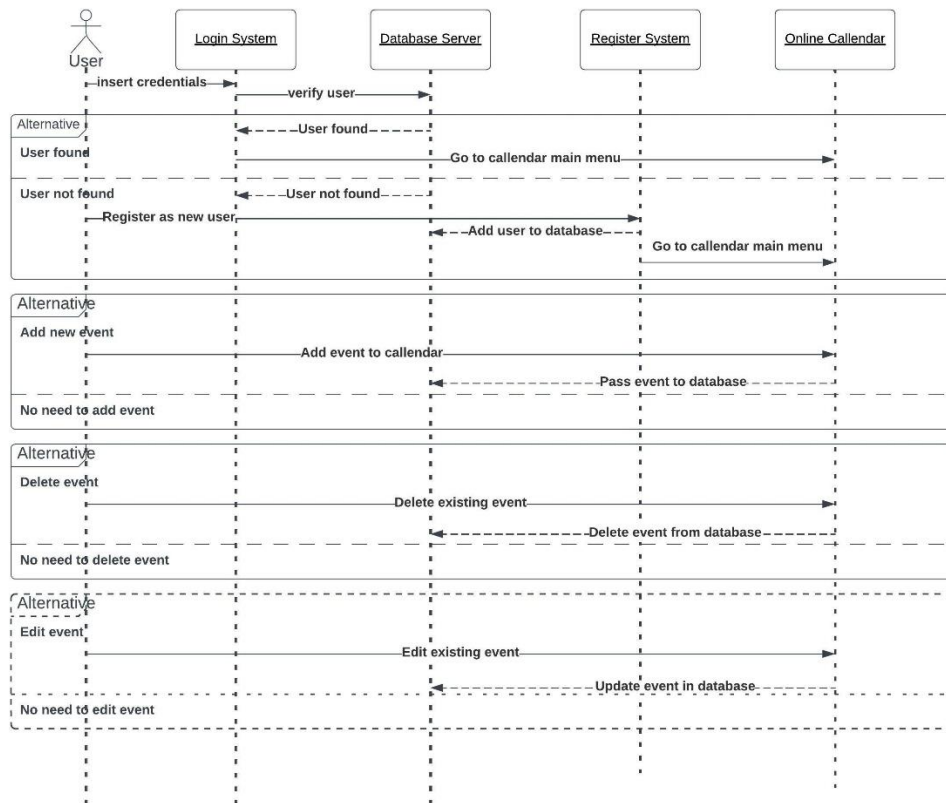


Διάγραμμα Συνεργασίας - Έκδοση 1

#### 4.1.5 Διαγράμματα Σειράς (1η έκδοση)

Το διάγραμμα σειράς αναπαριστά την ίδια πληροφορία με το διάγραμμα συνεργασίας. Η μόνη διαφορά είναι ότι ο χρόνος φαίνεται ρητά. Δηλώνεται με κατεύθυνση από πάνω προς τα κάτω η σειρά των μηνυμάτων που ανταλλάσσουν τα διάφορα αντικείμενα μεταξύ τους.

Στο παρακάτω διάγραμμα, απεικονίζεται το διάγραμμα σειράς για την φάση 4 της εφαρμογής. Ο χρήστης και τα τέσσερα αντικείμενα είναι οι δομές που αλληλοεπιδρούν μεταξύ τους μέσα στο σύστημα. Κάτω από μία τέτοια «δομή», με κάθετες διακεκομμένες γραμμές, φαίνεται η διάρκεια ζωής κάθε τέτοιου στοιχείου. Ο χρόνος κυλάει από πάνω προς τα κάτω. Επομένως τα μηνύματα που ανταλλάσσονται διαβάζονται από πάνω προς τα κάτω, σαν να εκτελούμε σάρωση στο διάγραμμα από την κορυφή προς τα κάτω.

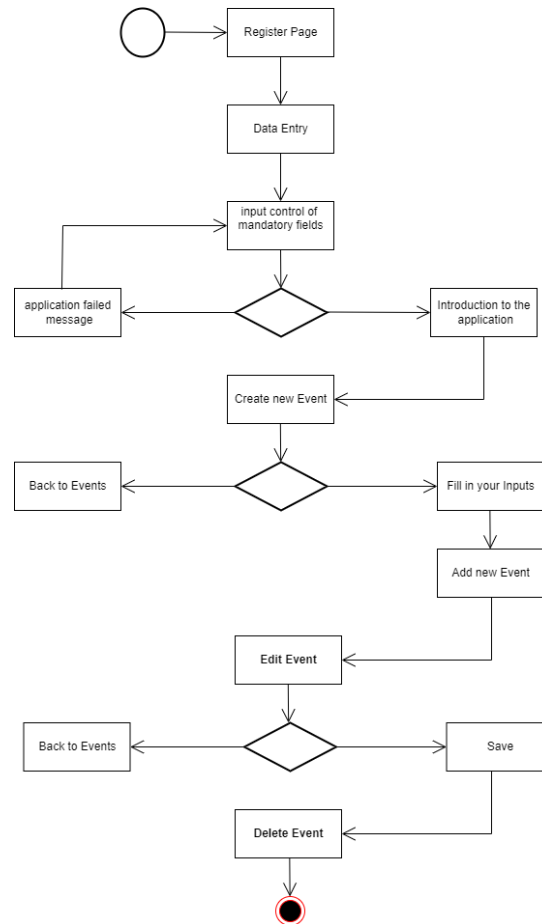


Διάγραμμα Σειράς - Έκδοση 1

#### 4.1.6 Διαγράμματα Δραστηριοτήτων (1η έκδοση)

Σε αυτό το Διάγραμμα Δραστηριοτήτων, αρχικά ο χρήστης συνδέεται στην αρχική σελίδα της εφαρμογής, ώστε να μεταβεί στη σελίδα σύνδεσης. Στη συνέχεια, ο χρήστης συμπληρώνει τα πεδία του ονόματος χρήστη και του κωδικού χρήστη όπως έχουν καταχωρηθεί στη φόρμα εγγραφής που είχε συμπληρώσει για να εισέλθει στην ιστοσελίδα. Αν ο χρήστης συμπληρώσει όλα τα υποχρεωτικά στοιχεία, τότε αποθηκεύεται στο σύστημα, διαφορετικά ο χρήστης μεταφέρεται ξανά στην σελίδα σύνδεσης για να εισάγει ξανά τα στοιχεία που του ζητήθηκαν. Σε περίπτωση που έχουν συμπληρωθεί τα στοιχεία του, ο χρήστης μεταβαίνει στην αρχική του σελίδα, διαφορετικά θα πρέπει να εισαγάγει ξανά τα στοιχεία του.

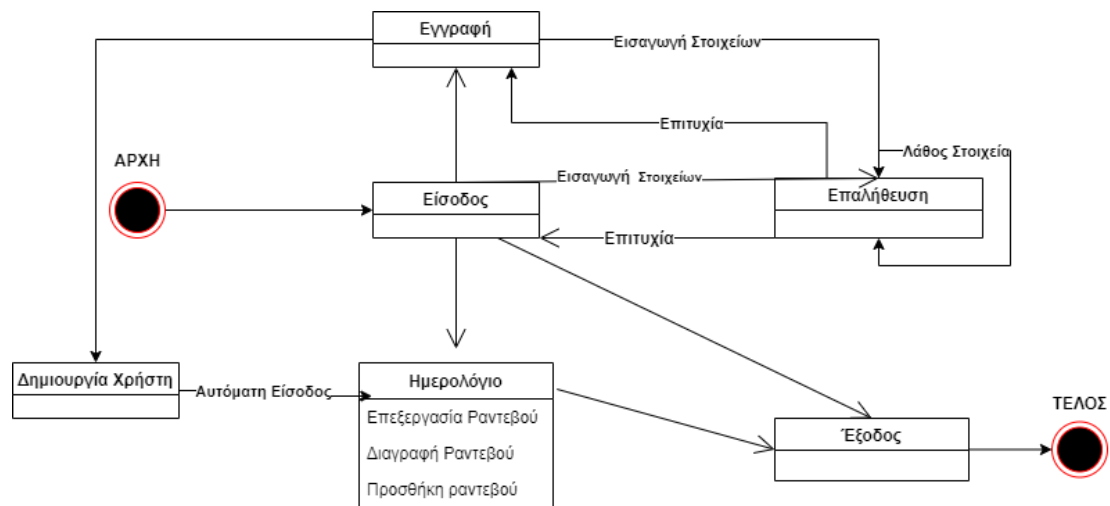
Μετάπειτα ο χρήστης έχει την δυνατότητα να φτιάξει ένα καινούργιο Event, όπου μπορεί να προσθέσει τα απαραίτητα στοιχεία για την δημιουργία αυτού του Event ή να πάει πίσω στην αρχική σελίδα. Ακολούθως, έχει την δυνατότητα να επεξεργαστεί ένα Event, η οποία επεξεργασία για να αποθηκευτεί ο χρήστης μπορεί είτε να πατήσει «αποθήκευση» είτε να επιστρέψει στην αρχική σελίδα. Τέλος δίνεται η δυνατότητα στον χρήστη να διαγράψει ένα Event.



Διάγραμμα Δραστηριοτήτων - Έκδοση 1

#### 4.1.7 Διαγράμματα Καταστάσεων (1η έκδοση)

Στο διάγραμμα αυτό αναπαρίσταται η ροή μεταξύ των καταστάσεων που περνάνε τα αιτήματα του χρήστη προς την εφαρμογή. Για παράδειγμα το αίτημα του χρήστη για είσοδο θα πρέπει πρώτα να περάσει από την επαλήθευση ώστε τα στοιχεία του να ελεγχθούν. Η προαναφερθείσα διαδικασία μπορεί να φανεί και από τα βέλη που συνδέουν τις καταστάσεις μεταξύ τους. Η χρονική σειρά φαίνεται από την φορά αυτών των βελών.



Διάγραμμα Καταστάσεων - Έκδοση 1

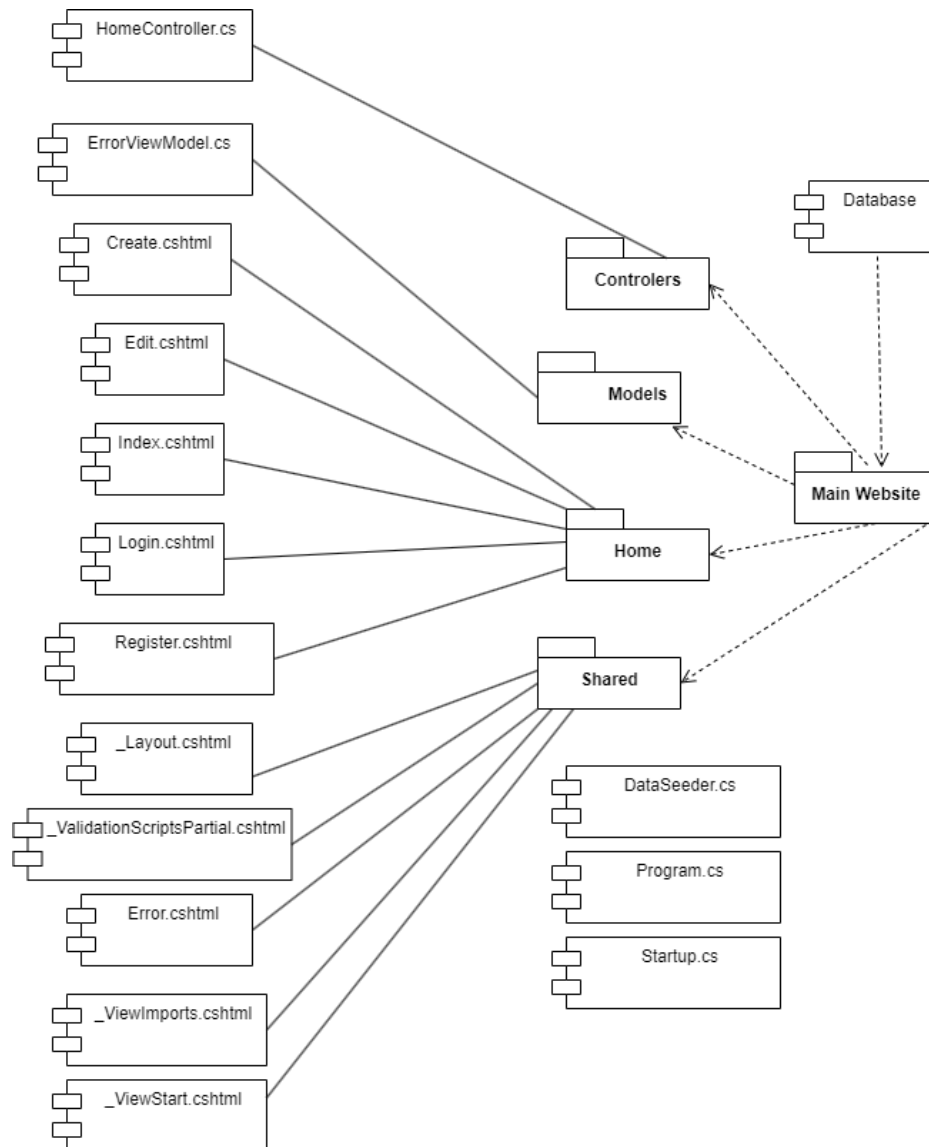
#### 4.1.8 Διαγράμματα Εξαρτημάτων (1η έκδοση)

Τα διαγράμματα εξαρτημάτων (Component Diagrams) μας δείχνουν τα συστατικά μέρη ενός συστήματος και τις εξαρτήσεις μεταξύ τους. Με τον όρο εξαρτήματα εννοούμε φυσικές μονάδες κώδικα (για παράδειγμα πηγαίο αρχείο, μια βιβλιοθήκη, εκτελέσιμο αρχείο κ.λπ.)

Η Εφαρμογή μας αποτελείται από τα παρακάτω τμήματα:

1. «HomeController»: Όταν η εφαρμογή ξεκινάει, αυτό το αρχείο το οποίο βρίσκεται στο φάκελο «Controllers» συνδέεται με την βάση και δίνει τις λειτουργικές δυνατότητες της εφαρμογής στον χρήστη.
2. Κάτω από τον φάκελο «Models» υπάρχει: «ErrorViewModel.cs»: Δίνει την δυνατότητα διαχείρισης σφαλμάτων στην εφαρμογή
3. Κάτω από τον φάκελο «Home» υπάρχουν τα παρακάτω (δυνατότητες του χρήστη):
4. Όταν ο χρήστης εγγράφεται στο σύστημα έχει πρόσβαση στα παρακάτω:
  - «login.cshtml»: Η σελίδα με την οποία ο χρήστης εισάγει το «Username» και το «Password» του για να συνδεθεί.
  - «Create.cshtml»: Δημιουργία καινούργιου Event
  - «Edit.cshtml»: Επεξεργασία υπάρχοντος Event
  - «Index.cshtml»: Διαγραφή ενός Event.
5. Στον κοινόχρηστο φάκελο («Shared») υπάρχουν:
  - «\_ValidationScriptsPartial.cshtml»
  - «\_ViewImports.cshtml »:

- ο « \_ViewStarts.cshtml »: Επιτρέπει στον χρήστη να δει την επιφάνεια της εφαρμογής
- ο « Error.cshtml»: Διαχειρίζεται τα λάθη του συστήματος
- ο « Layout.cshtml»: Η εξωτερική εμφάνιση της εφαρμογής

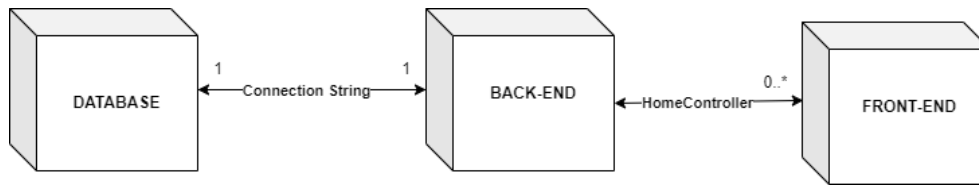


Διάγραμμα Εξαρτημάτων - Έκδοση 1

#### 4.1.9 Διαγράμματα Διανομής (1η έκδοση)

Σε κανονικές συνθήκες αυτό το διάγραμμα δείχνει τη σύνδεση του υλικού σε μία εφαρμογή. Στην περίπτωσή μας όμως, καθώς η εφαρμογή μας δεν διαθέτει υλικό, θα υποθέσουμε πως αποτελείται από 3 κομμάτια το Front και Back End της σελίδας μας και την βάση δεδομένων μας. Τα κομμάτια αυτά συνδέονται μεταξύ τους είτε με την βοήθεια Connection String (DB→BACK END) είτε με την βοήθεια μιας κλάσης HomeController (BACK END → FRONT END). Οι φορές των βελών δείχνουν την αλληλεπίδραση, καθώς είναι διπλής

κατεύθυνσης και οι αριθμοί δείχνουν με πόσα κομμάτια επικοινωνούν. Για παράδειγμα, το BACK END μπορεί να επικοινωνεί με πολλές από τις cshtml σελίδες οπότε γράφουμε 0..\* (καμία ή άπειρες).



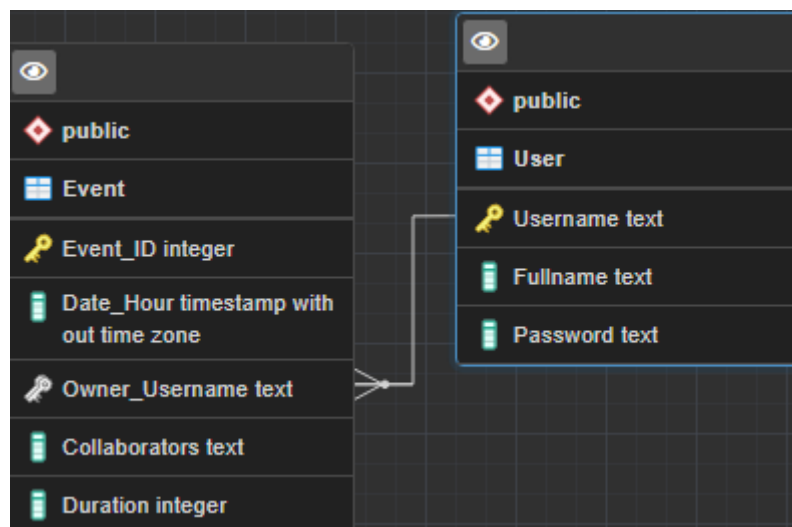
Διάγραμμα Διανομής - Φάση 1

## 4.2 Υλοποίηση-Έλεγχος

### 4.2.1 Υλοποίηση: 1η εκτελέσιμη έκδοση

Στην πρώτη εκτελέσιμη έκδοση του λογισμικού συμπεριλαμβάνονται μεταξύ άλλων τα ακόλουθα:

κατασκευή βάσης δεδομένων που καταγράφονται τα ραντεβού και οι χρήστες της εφαρμογής. Το διάγραμμα οντοτήτων σχέσεων της βάσης είναι το ακόλουθο:



Διάγραμμα οντοτήτων/σχέσεων της βάσης δεδομένων - Έκδοση 1

Σύστημα σύνδεσης και εγγραφής ενός χρήστη στην πλατφόρμα: κατά την σύνδεση ενός χρήστη (αφού εισάγει το username και το password του) διασταυρώνονται τα στοιχεία του χρήστη με τα δεδομένα της βάσης δεδομένων. Εάν ο χρήστης υπάρχει, τότε η σύνδεση είναι επιτυχής, ενώ ζητείται η εγγραφή του χρήστη ή η επανεισαγωγή των στοιχείων του σε κάθε άλλη περίπτωση.

Στο κεντρικό μενού της εφαρμογής, ο χρήστης μπορεί να παρακολουθεί όλα τα διαθέσιμα ραντεβού, τα οποία έχει δημιουργήσει αυτός.

Εφόσον θελήσει ο χρήστης, υπάρχει η δυνατότητα διαγραφής ενός δικού του ραντεβού, τρέχοντας ένα ερώτημα διαγραφής στη βάση δεδομένων.

Για την εισαγωγή ενός νέου ραντεβού, ο χρήστης μεταφέρεται σε μία καινούρια σελίδα, όπου υπό μορφή φόρμας ζητούνται τα απαραίτητα στοιχεία (ημερομηνία, συνεργάτες και διάρκεια) για την δημιουργία του ραντεβού. Έπειτα, εκτελείται ένα insert query στη βάση, για να προστεθεί αυτό το νέο ραντεβού σε αυτή.

Με ανάλογο τρόπο γίνεται και η τροποποίηση ενός ραντεβού (αντί για insert query έχουμε update query).

#### 4.2.2 Αναφορά ελέγχου για την 1η εκτελέσιμη έκδοση

Όσον αφορά το λογισμικό που έχει ήδη κατασκευαστεί, παρά το γεγονός ότι η κατασκευή του βασίζεται και θεμελιώνεται πάνω στις λειτουργικές απαιτήσεις που αναφέρθηκαν σε προηγούμενη ενότητα, το πρόγραμμα παρουσιάζει ορισμένες ελλείψεις που χρίζουν βελτίωσης και αναφέρονται παρακάτω:

Ο χρήστης πρέπει να αποσυνδέεται από το σύστημα.

Στην εφαρμογή μπορεί να ορίζονται συνεργάτες σε ένα event και μη χρήστες της συγκεκριμένης πλατφόρμας. Ωστόσο, για την δημιουργία ενός ραντεβού είναι απαραίτητη η δημιουργία λογαριασμού σε αυτή.

Πρέπει να υποστηρίζεται ένα σύστημα ειδοποιήσεων. Σύμφωνα με τις απαιτήσεις της εφαρμογής, για κάθε δημιουργία, διαγραφή ή τροποποίηση ενός ραντεβού, θα πρέπει να ενημερώνονται οι αντίστοιχοι χρήστες (μόνο αυτοί που είναι χρήστες της πλατφόρμας).

## 5. Φάση: Κατασκευή (Construction)

### 5.1 Ανάλυση-Σχεδιασμός

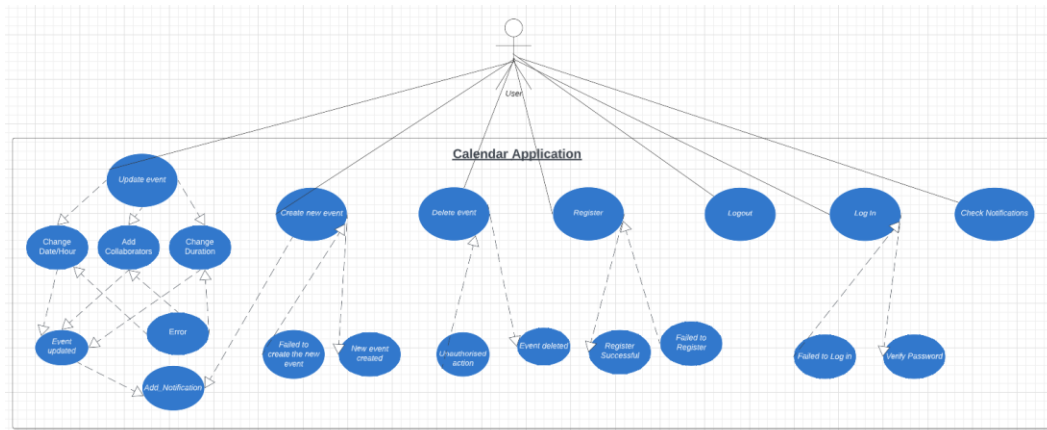
#### 5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3η έκδοση)

Στη τελευταία έκδοση του διαγράμματος περιπτώσεων χρήσης αλλάξαμε/προσθέσαμε τα παρακάτω:

- αρχικά προσθέσαμε την λειτουργία, που δίνεται απευθείας στο χρήστη μέσω της εφαρμογής να μπορεί να βλέπει τυχόν ειδοποιήσεις που του έρχονται από νέα ραντεβού.
- επίσης προσθέσαμε το κουμπί “Add\_Notification”, το οποίο υποδηλώνει πως μετά



από την δημιουργία ή αλλαγή κάποιου ραντεβού μπορεί να υπάρξει τυχόν ειδοποίηση σε κάποιον χρήστη που πρέπει να παραβρεθεί στο ραντεβού αυτό.

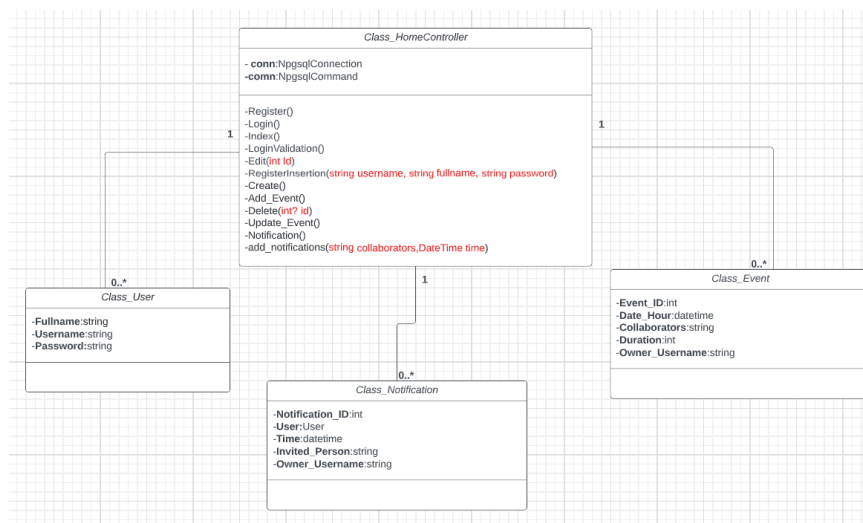


Διάγραμμα Περιπτώσεων Χρήσης - Έκδοση 3

### 5.1.2 Διαγράμματα Τάξεων (3η έκδοση)

Στην τελευταία έκδοση του διαγράμματος τάξεων αλλάξαμε/προσθέσαμε τα εξής:

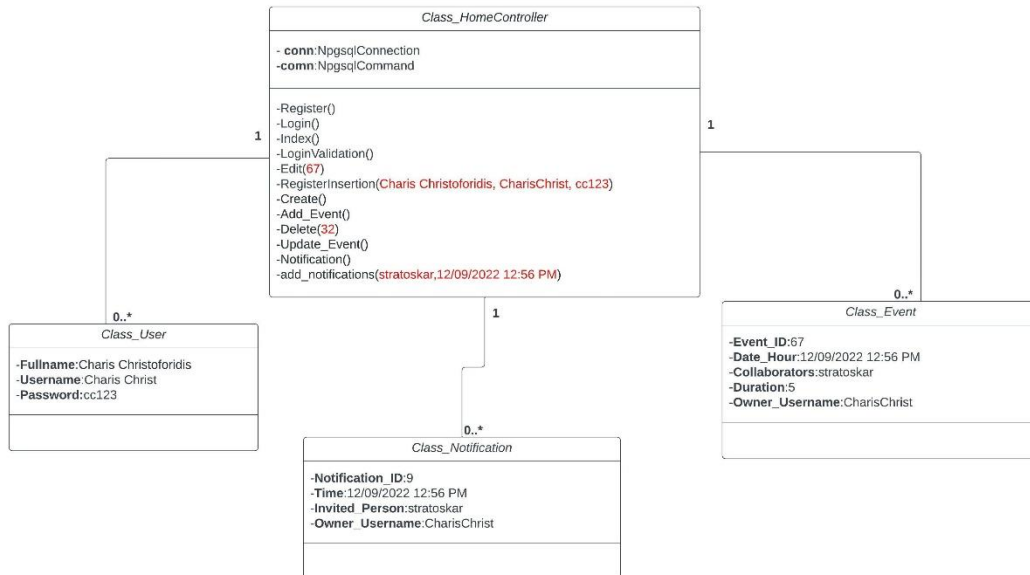
- προσθέσαμε την κλάση “Notification” με τα χαρακτηριστικά της, η οποία συνδέεται με την κλάση “HomeController”, καθώς αυτή είναι που δημιουργεί και αντικείμενα τύπου ειδοποίησης.
- αλλάξαμε τον τύπο του πεδίου Owner\_Username σε String και του DateHour σε DateTime στην κλάση Event.
- προσθέσαμε τις συναρτήσεις Notification() και addNotification() στην κλάση HomeController.



Διάγραμμα Τάξεων - Έκδοση 3

### 5.1.3 Διαγράμματα Αντικειμένων (2η έκδοση)

Το διάγραμμα αντικειμένων για μία δεδομένη χρονική στιγμή εκτέλεσης του προγράμματος είναι το ακόλουθο:



Διάγραμμα Αντικειμένων - Έκδοση 2

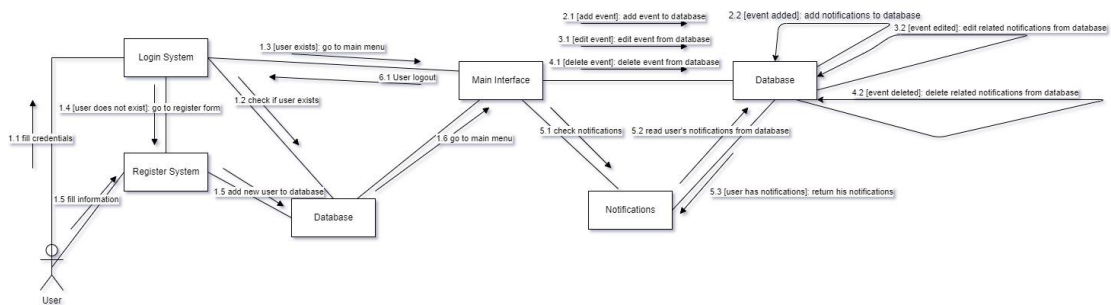
Συγκεκριμένα, στο συγκεκριμένο διάγραμμα γίνεται εγγραφή ενός νέου χρήστη. Ταυτόχρονα, ο καινούριος αυτός χρήστης δημιουργεί ένα νέο ραντεβού στην πλατφόρμα, ορίζοντας και έναν συνεργάτη. Τέλος, δημιουργείται μία νέα ειδοποίηση προς τον συνεργάτη που ορίστηκε στο ραντεβού αυτό.

### 5.1.4 Διαγράμματα Συνεργασίας (2η έκδοση)

Το διάγραμμα συνεργασίας της 5ης φάσης είναι σχεδόν το ίδιο με το διάγραμμα συνεργασίας της προηγούμενης έκδοσης. Οι αλλαγές που εισήχθησαν είναι οι ακόλουθες:

- ο χρήστης πλέον με εντολή του μπορεί να δει τις ειδοποιήσεις του.
- η όποια αλλαγή, διαγραφή ή δημιουργία ενός ραντεβού, επηρεάζει τις ειδοποιήσεις του συστήματος, πράγμα που αποτυπώνεται με βελάκια ανάδρασης από και προς την βάση δεδομένων.
- ταυτόχρονα, φαίνεται λεπτομερέστερα η διαδικασία σύνδεσης και αποσύνδεσης του χρήστη από το σύστημα.
- επιπλέον, δηλώνεται ρητά στο διάγραμμα ότι η αποτυχία ταυτοποίησης του χρήστη τον οδηγεί στην εγγραφή του μέσω της φόρμας register.

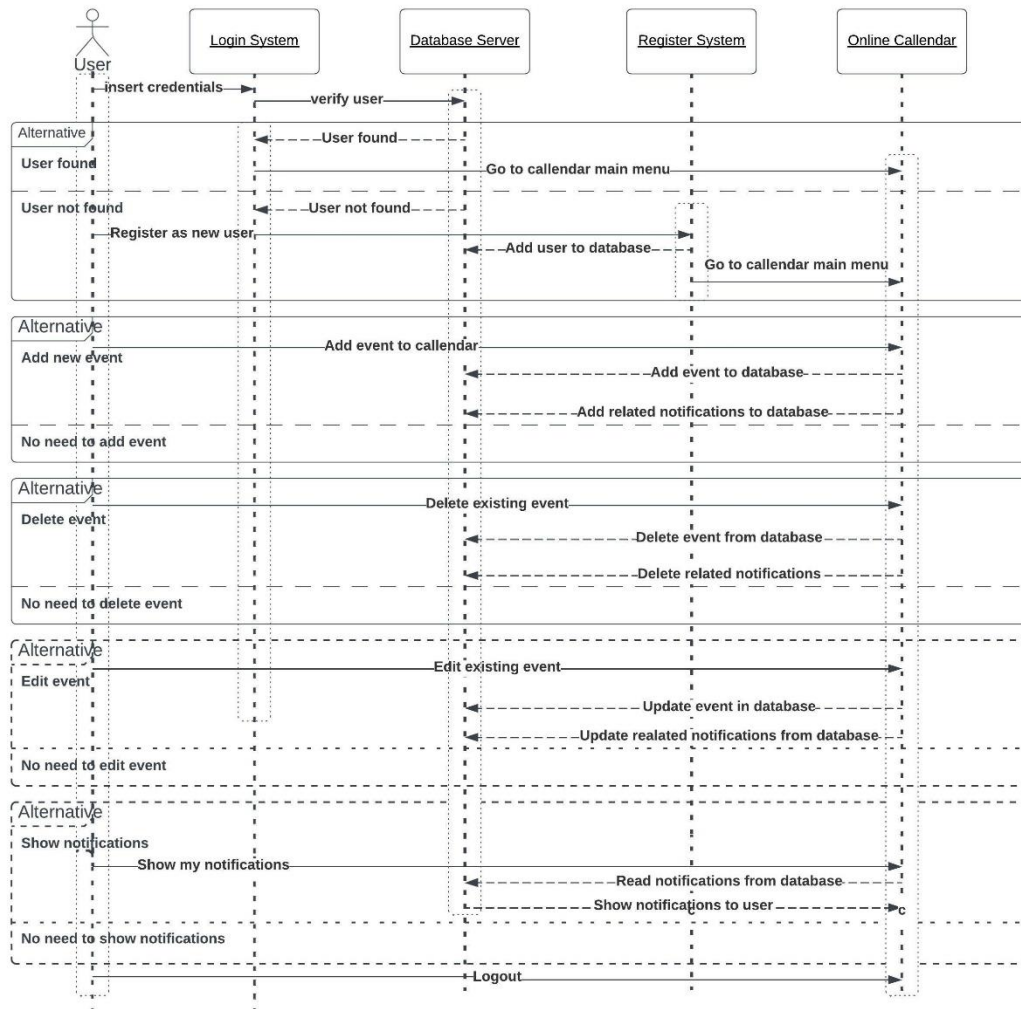
Το νέο διάγραμμα συνεργασίας είναι το ακόλουθο και λειτουργεί με παρόμοιο τρόπο, όπως περιεγράφηκε σε προηγούμενη ενότητα Διαγράμματα Συνεργασίας (1η έκδοση):



Διάγραμμα Συνεργασίας - Έκδοση 2

### 5.1.5 Διαγράμματα Σειράς (2η έκδοση)

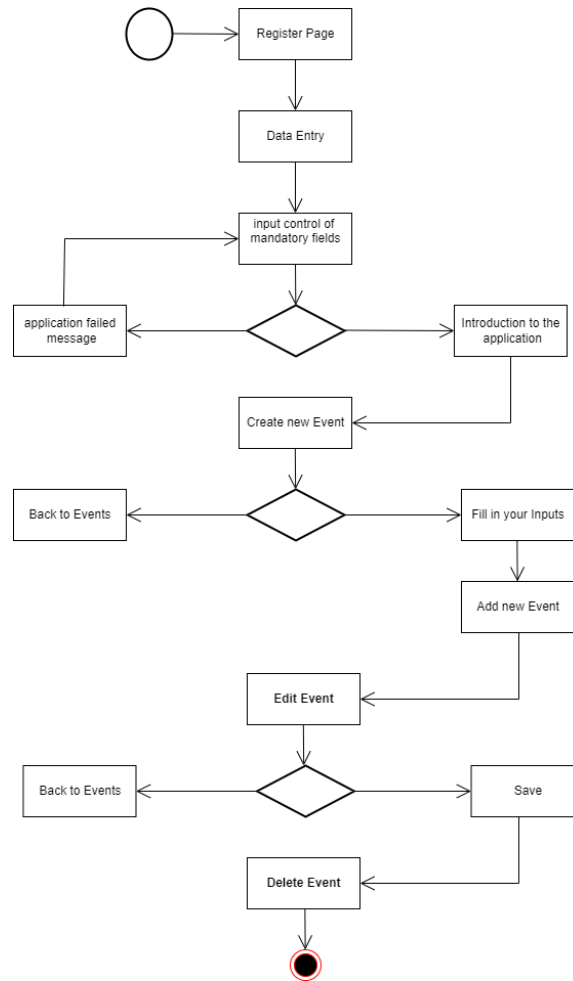
Το διάγραμμα σειράς ακολουθεί με τη σειρά του τις τροποποιήσεις που εισήχθησαν στο διάγραμμα συνεργασίας. Στο διάγραμμα σειράς αποτυπώνεται ρητά ο χρόνος. Παρακάτω, φαίνεται το διάγραμμα σειράς της νέας έκδοσης:



Διάγραμμα Σειράς - Έκδοση 2

### 5.1.6 Διαγράμματα Δραστηριοτήτων (2η έκδοση)

Κατά την δεύτερη αναθεώρηση του διαγράμματος δραστηριοτήτων δεν αλλάξαμε κάτι. Το διάγραμμα παραμένει το ίδιο, σε σχέση με αυτό της προηγούμενης έκδοσης καθώς οι ειδοποιήσεις που λαμβάνει ο χρήστης από την πλατφόρμα δεν αποτελούν δραστηριότητες.

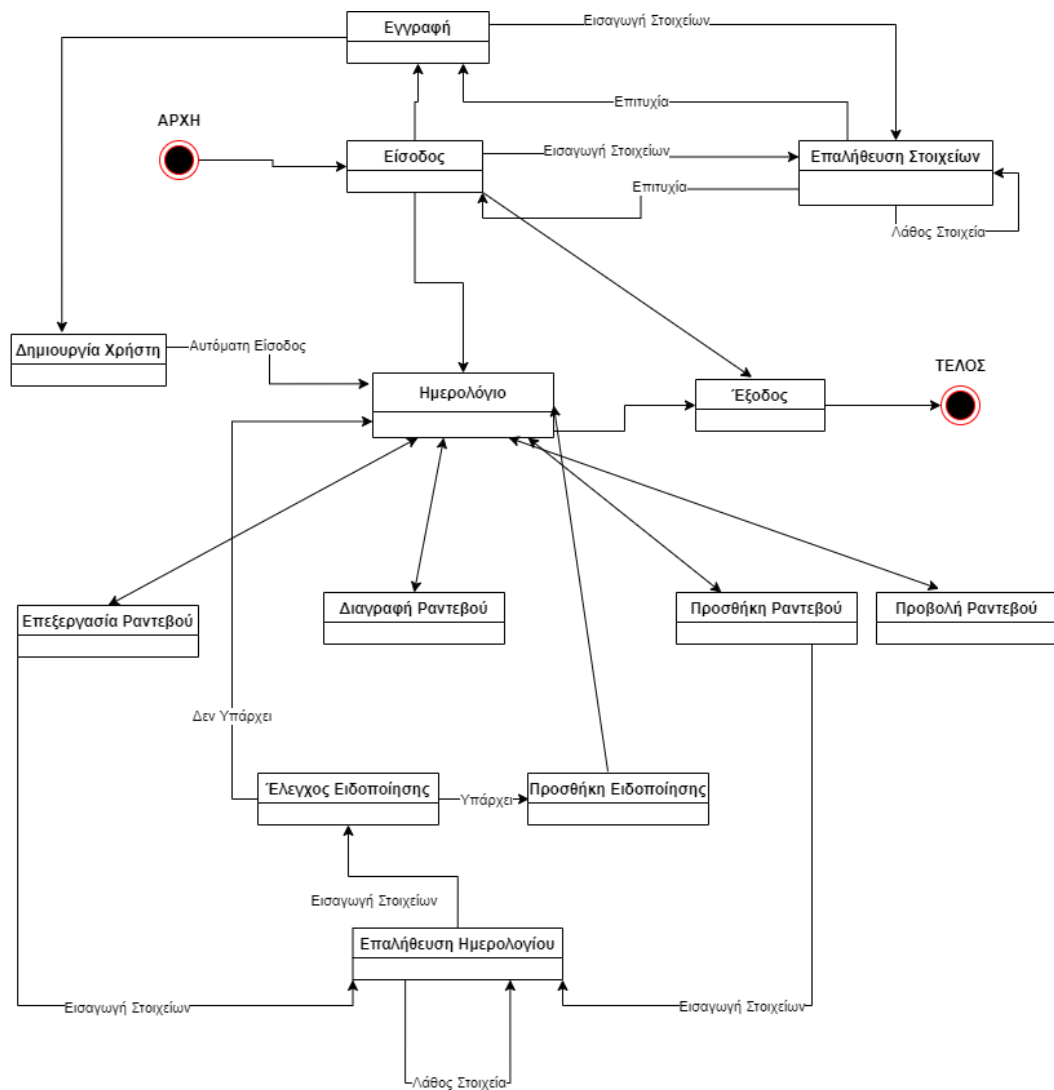


Διάγραμμα Δραστηριοτήτων - Έκδοση 2

### 5.1.7 Διαγράμματα Καταστάσεων (2η έκδοση)

Στην τελευταία έκδοση του διαγράμματος καταστάσεων προσθέσαμε/αλλάξαμε τα εξής:

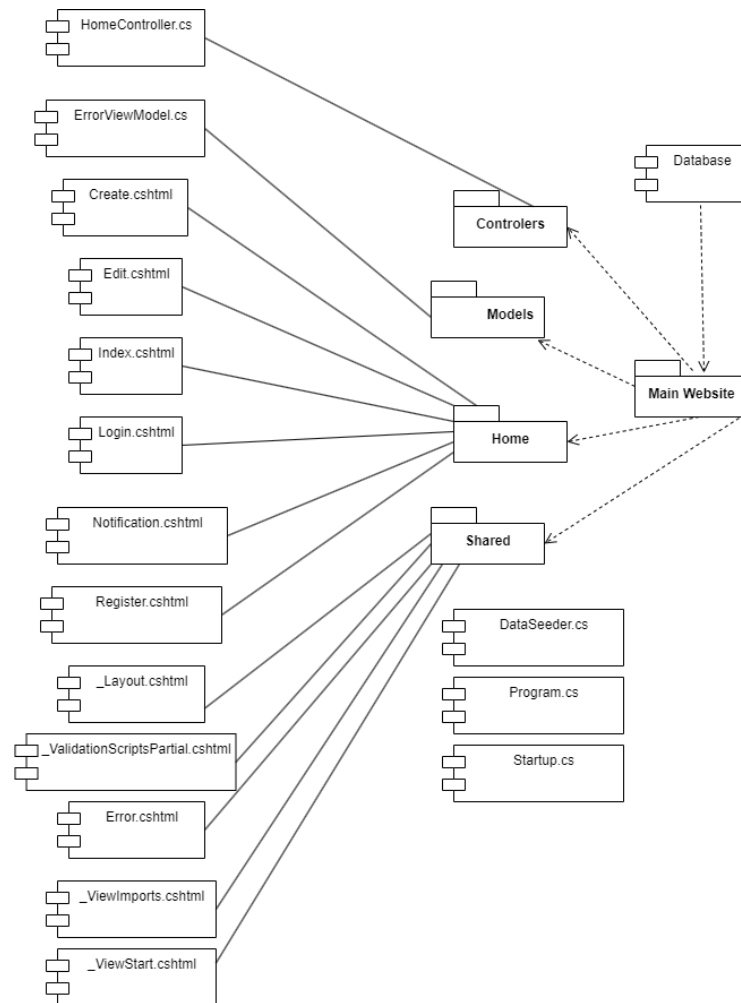
- αναπτύχθηκαν οι λειτουργίες της διαγραφής, εισαγωγής και επεξεργασίας ενός ραντεβού, καθώς και οι έλεγχοι, οι οποίοι γίνονται για να εγκυροποιήσουν τέτοιες ενέργειες.
- επίσης προσθέσαμε και τον τρόπο λειτουργίας των Notifications, τα οποία έχουν άμεση σχέση με την δημιουργία και επεξεργασία των ραντεβού, αφού χωρίς αυτά δεν μπορούν να υπάρξουν και τα notifications αυτά κάθε αυτά.



Διάγραμμα Καταστάσεων - Έκδοση 2

### 5.1.8 Διαγράμματα Εξαρτημάτων (2η έκδοση)

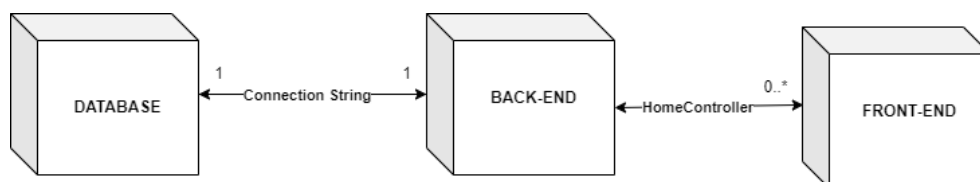
Η δεύτερη έκδοση του διαγράμματος εξαρτημάτων περιλαμβάνει μεταξύ των αρχείων του λογισμικού και το Notification.cshtml, την σελίδα στην οποία φαίνονται οι ειδοποιήσεις στο τον χρήστη. Κατά τα άλλα, το διάγραμμα αυτό παραμένει σχεδόν άθικτο με αυτό της προηγούμενης έκδοσης.



Διάγραμμα Εξαρτημάτων - Έκδοση 2

### 5.1.9 Διαγράμματα Διανομής (2η έκδοση)

Στην τελευταία έκδοση του διαγράμματος διανομής δεν αλλάξαμε τίποτα καθώς τα «υλικά» της εφαρμογής μας παρέμειναν τα ίδια από την προηγούμενη φάση.



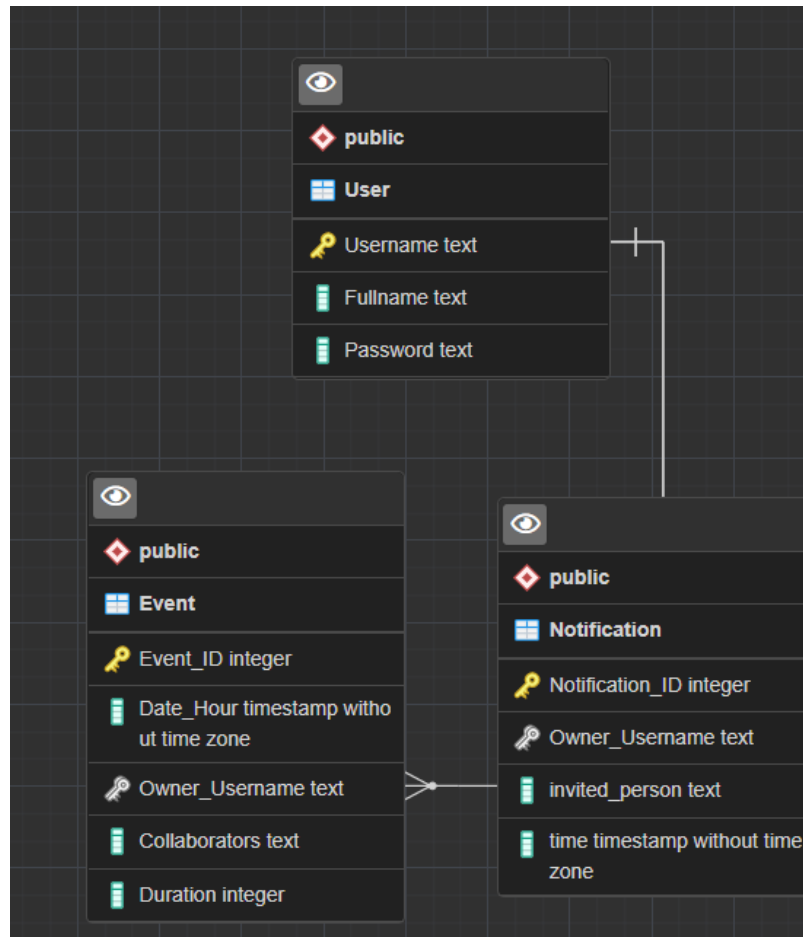
Διάγραμμα Διανομής - Έκδοση 2

## 5.2 Υλοποίηση-Έλεγχος

### 5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση

Στην τελική εκτελέσιμη έκδοση του λογισμικού συμπεριλαμβάνονται τα ακόλουθα:

- Κατασκευή βάσης δεδομένων που καταγράφονται τα ραντεβού και οι χρήστες της εφαρμογής καθώς και οι ειδοποιήσεις που λαμβάνουν οι χρήστες για τα επερχόμενα ραντεβού τους. Το διάγραμμα οντοτήτων σχέσεων της βάσης είναι το ακόλουθο:



Διάγραμμα οντοτήτων/σχέσεων της βάσης δεδομένων - Έκδοση 2

- Σύστημα σύνδεσης, αποσύνδεσης και εγγραφής ενός χρήστη στην πλατφόρμα: κατά την σύνδεση ενός χρήστη (αφού εισάγει το username και το password του) διασταυρώνονται τα στοιχεία του χρήστη με τα δεδομένα της βάσης δεδομένων. Εάν ο χρήστης υπάρχει, τότε η σύνδεση είναι επιτυχής, ενώ ζητείται η εγγραφή του χρήστη ή η επανεισαγωγή των στοιχείων του σε κάθε άλλη περίπτωση. Τέλος αφού ο χρήστης συνδεθεί μπορεί οποιαδήποτε στιγμή να αποσυνδεθεί από την εφαρμογή με την χρήση του σχετικού κουμπιού.
- Στο κεντρικό μενού της εφαρμογής(ημερολόγιο), ο χρήστης μπορεί να παρακολουθεί όλα τα διαθέσιμα ραντεβού, τα οποία έχει δημιουργήσει αυτός. Καθώς και να ειδοποιηθεί για επερχόμενα ραντεβού που έχει κληθεί να συμμετάσχει από άλλους χρήστες.
- Εφόσον θελήσει ο χρήστης, υπάρχει η δυνατότητα διαγραφής ενός δικού του



ραντεβού, τρέχοντας ένα ερώτημα διαγραφής στη βάση δεδομένων.

- Για την εισαγωγή ενός νέου ραντεβού, ο χρήστης μεταφέρεται σε μία καινούρια σελίδα, όπου υπό μορφή φόρμας ζητούνται τα απαραίτητα στοιχεία (ημερομηνία, συνεργάτες και διάρκεια) για την δημιουργία του ραντεβού. Έπειτα, εκτελείται ένα insert query στη βάση, για να προστεθεί αυτό το νέο ραντεβού σε αυτή.
- Με ανάλογο τρόπο γίνεται και η τροποποίηση ενός ραντεβού (αντί για insert query έχουμε update query), αφού ο χρήστης αποφασίσει ποιο πεδίο του ραντεβού (μπορεί και όλα) επιθυμεί να αλλάξει. Το μόνο πεδίο που δεν διατίθεται στο χρήστη για αλλαγή είναι το πεδίο του ιδιοκτήτη του ραντεβού, καθώς αυτό είναι ο ίδιος ο χρήστης.
- Για την δημιουργία των ειδοποιήσεων που πρέπει να εμφανίζονται στο χρήστη, υπάρχει ειδικό κουμπί στην αρχική σελίδα του ημερολογίου. Ο χρήστης επιλέγοντάς το μεταφέρεται στην σελίδα ειδοποιήσεων του που μπορεί να διαβάσει τις ειδοποιήσεις του και να απαντήσει θετικά ή αρνητικά προς αυτές. Μία ειδοποίηση δημιουργείται στη βάση όταν κάποιος άλλος χρήστης τον προσθέσει ως συμμετέχοντα σε κάποιο ραντεβού είτε κατά τη διάρκεια της δημιουργίας είτε κατά τη διάρκεια της επεξεργασίας κάποιου ραντεβού. Αφού ο χρήστης δώσει απάντηση θετική/αρνητική στην ειδοποίηση η ίδια σταματάει να υπάρχει και διαγράφεται και από την βάση καθώς εκτέλεσε τον σκοπό της.

## 5.2.2 Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση

Όσον αφορά το λογισμικό που έχει κατασκευαστεί, βασίζεται και θεμελιώνεται πάνω στις λειτουργικές απαιτήσεις που αναφέρθηκαν στις προηγούμενες ενότητες, το πρόγραμμα θεωρείται πλήρες καθώς καλύψαμε τις παρακάτω ελλείψεις της προηγούμενης φάσης:

- ο χρήστης πρέπει να αποσυνδέεται από το σύστημα.
- στην εφαρμογή μπορεί να ορίζονται συνεργάτες σε ένα ραντεβού και μη χρήστες της συγκεκριμένης πλατφόρμας. Ωστόσο, για την δημιουργία ενός ραντεβού είναι απαραίτητη η δημιουργία λογαριασμού σε αυτή.
- πρέπει να υποστηρίζεται ένα σύστημα ειδοποιήσεων. Σύμφωνα με τις απαιτήσεις της εφαρμογής, για κάθε δημιουργία, διαγραφή ή τροποποίηση ενός ραντεβού, θα πρέπει να ενημερώνονται οι αντίστοιχοι χρήστες (μόνο αυτοί που είναι χρήστες της πλατφόρμας).

Έχοντας καλύψει τα παραπάνω νιώθουμε αρκετά ικανοποιημένοι με την τελική μορφή του λογισμικού μας ,καθώς θα ήταν κάτι που θα χρησιμοποιούσαμε και οι ίδιοι στην καθημερινότητά μας. Στον χρήστη παρέχονται όλες οι απαραίτητες λειτουργίες που είχαμε αναφέρει στις πρώτες φάσεις τις εργασίας μας.

## 6. Εγχειρίδιο Χρήστη

### 6.1 Σύντομη παρουσίαση του προγράμματος

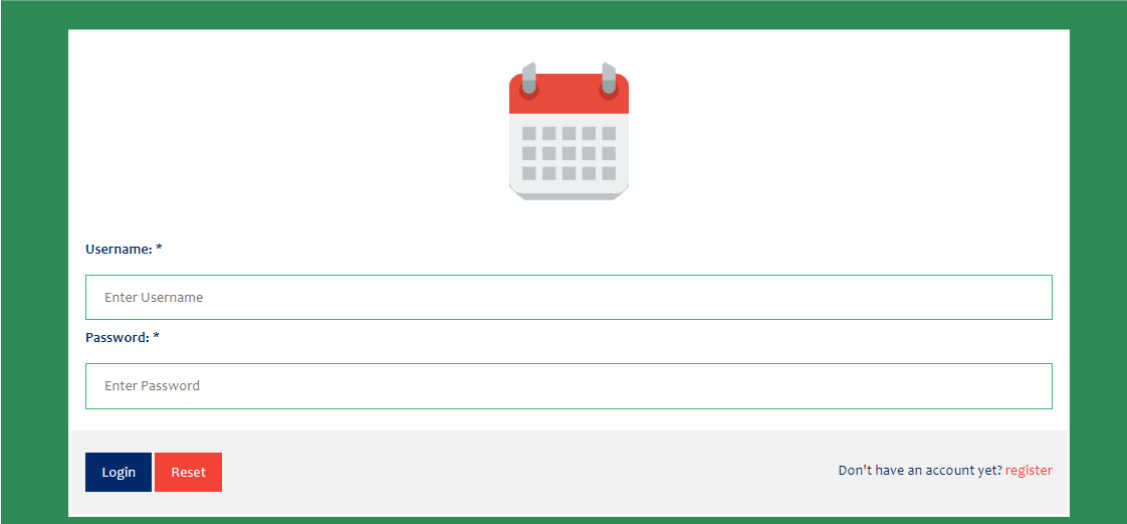
#### Τεχνικά Στοιχεία:

Το εν λόγω πρόγραμμα αποτελεί μία διαδικτυακή εφαρμογή, η οποία είναι δομημένη ως MVC project (Model – View – Controller) σε ASP .NET 5 σε γλώσσα C#.

#### Οδηγίες λειτουργίας του λογισμικού:

Κατά την εκτέλεση του προγράμματος, θα μεταφερθείτε στην σελίδα που υπάρχει η φόρμα σύνδεσης της πλατφόρμας (Login Form), στην οποία εισάγετε το Username και τον κωδικό σας.

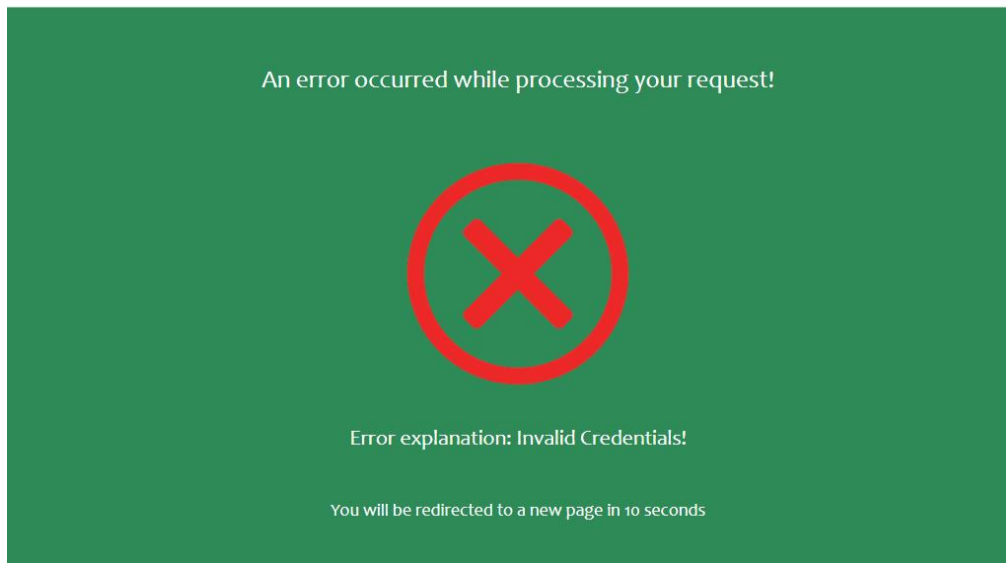
Online Callendar



*Login Form*

Σε περίπτωση εισαγωγής λανθασμένων στοιχείων, εμφανίζεται μία σελίδα σφάλματος που σας εξηγεί τι ακριβώς πήγε στραβά.

Online Callendar



*Μήνυμα Σφάλματος προς τον Χρήστη*

Επιπρόσθετα, σε περίπτωση που δεν έχετε ήδη λογαριασμό στην πλατφόρμα, μπορείτε να μεταβείτε στην σελίδα για νέα εγγραφή χρήστη, πατώντας τον σχετικό σύνδεσμο «register» που υπάρχει κατά την εκτέλεση του προγράμματος.

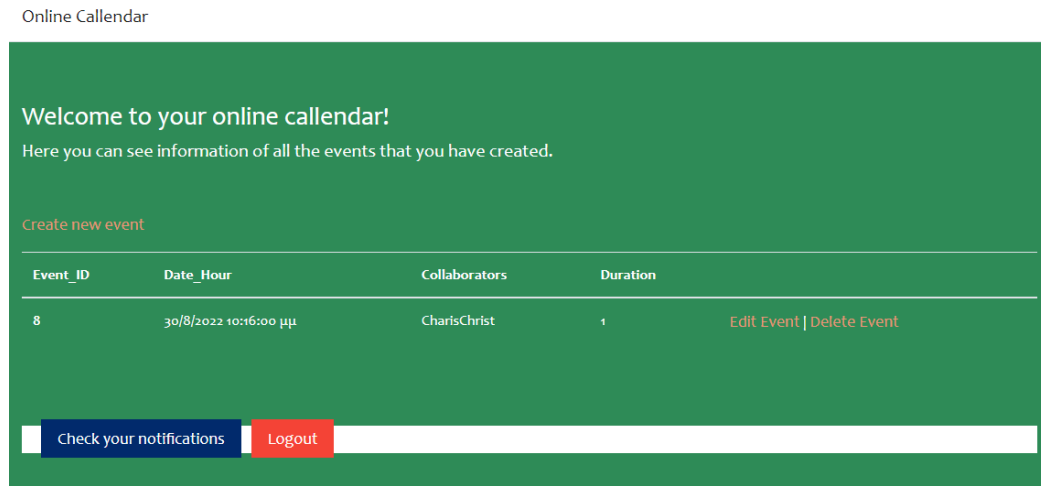
Online Callendar

A screenshot of a web application interface with a green border. At the top center is a calendar icon. Below it, the text "Hello user, create your account:" is displayed. There are three input fields: "Full name: \*" with a placeholder "Enter your fullname", "Username: \*" with a placeholder "Enter your username", and "Password: \*" with a placeholder "Enter your password". At the bottom left are two buttons: "Register" (blue) and "Reset" (red). At the bottom right, there is a link that says "Already have an account? login".

*Σελίδα Register*

Σε συνέχεια της επιτυχούς σύνδεσής σας, θα σας εμφανιστεί στην οθόνη σας η κεντρική σελίδα (menu), στην οποία φαίνονται όλα τα ραντεβού, τα οποία έχετε

δημιουργήσει. Στο κεντρικό αυτό μενού της εφαρμογής μπορείτε να επέμβετε στο ημερολόγιό σας προκαλώντας αλλαγές σε αυτό. Με άλλα λόγια, έχετε την δυνατότητα να πατήσετε τις σχετικές επιλογές για εισαγωγή, τροποποίηση ή διαγραφή ενός ραντεβού. Ταυτόχρονα, υπάρχουν διαθέσιμες επιλογές για προβολή των ειδοποιήσεών σας, δηλαδή προσκλήσεις σε ραντεβού που έχετε δεχτεί από άλλους χρήστες της πλατφόρμας, καθώς και για έξοδό σας από το σύστημα (logout).



Κεντρικό Μενού της Εφαρμογής

Η διαδικασία για εισαγωγή ενός νέου ραντεβού γίνεται ως εξής, αφού πατήσετε τον σχετικό σύνδεσμο από το μενού, μεταφέρεστε σε μία νέα σελίδα όπου χρειάζεται να δηλώσετε τα χαρακτηριστικά του καινούριου ραντεβού σας (ώρα και μέρα, τους συνεργάτες σας χωρισμένους με κόμμα και την διάρκεια του ραντεβού σας). Έπειτα, πατάτε την επιλογή «Create» για δημιουργία του ραντεβού σας. Οι συνεργάτες, αν υπάρχουν, θα λάβουν ειδοποιήσεις για αυτό το νέο ραντεβού.

### Online Callendar

**Add a new event**

Date\_Hour

mm/dd/yyyy --:-- --

Collaborators

Duration

Create

[Go back to your events](#)

*Φόρμα Εισαγωγής Νέου Ραντεβού*

Η διαδικασία της τροποποίησης ενός ραντεβού περιλαμβάνει παρόμοια διαδικασία με αυτή της δημιουργίας ενός νέου ραντεβού. Κατά την τροποποίηση ενός ραντεβού, μπορείτε να αλλάξετε είτε την ώρα, είτε την ημέρα, είτε τους συνεργάτες, είτε την διάρκειά του. Για επικύρωση των αλλαγών σας θα πρέπει να απαντήσετε καταφατικά στο μήνυμα ειδοποίησης που «στέλνει» η εφαρμογή.

localhost:44394 says  
Are you sure you want to update this event?

OK Cancel

Online Calendar

Edit this Event

Date\_Hour  
08/30/2022 10:16 PM

Collaborators  
CharisChrist

Duration  
1

Save

Go back to your events

*Τροποποίηση Συγκεκριμένου Ραντεβού*

Για την διαγραφή του ραντεβού θα χρειαστεί να απαντήσετε καταφατικά στο προειδοποιητικό μήνυμα του συστήματος.

localhost:44394 says

Are you sure you want to delete this event?

OK

Cancel

*Επιβεβαίωση Διαγραφής Ραντεβού*

Τέλος, όσον αφορά τις ειδοποιήσεις σας, σας δίνεται η επιλογή να απαντήσετε σε αυτές δηλώνοντας εάν να παρευρεθείτε στο ραντεβού ή όχι.

Online Calendar

Here are your current notifications:

Notification_ID	Owner_Username	time	
26	CharisChrist	30/9/2022 12:48:00 πμ	Accept   Decline
27	CharisChrist	29/9/2022 3:48:00 πμ	Accept   Decline

Go back to your events

*Διαχείριση των Ειδοποιήσεων*

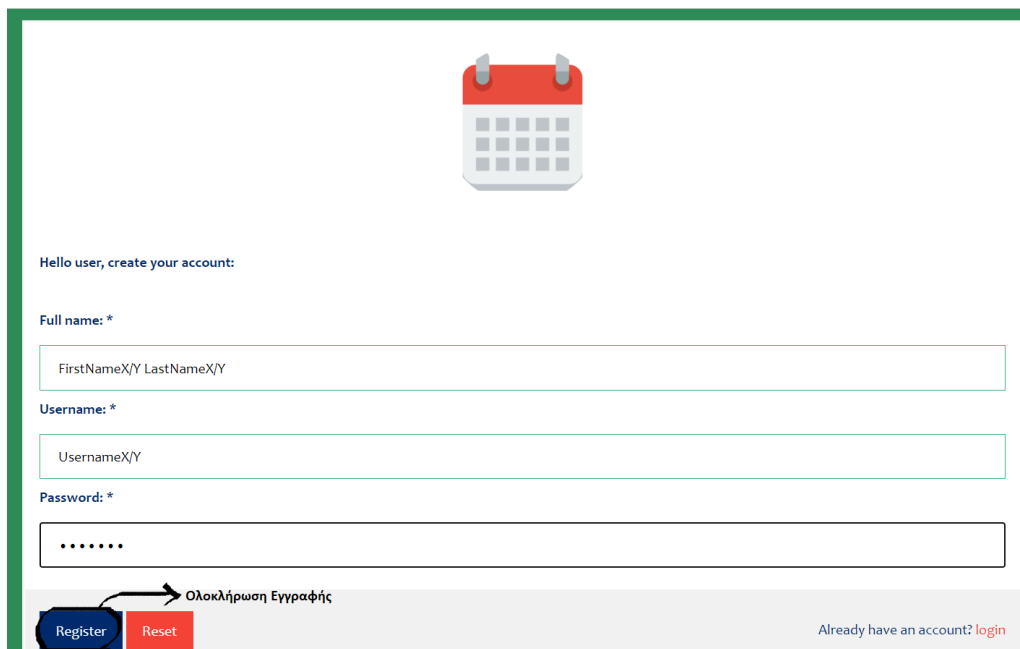
## 6.2 Παρουσίαση σεναρίων λειτουργίας

Στην παρούσα φάση της εργασίας μας θα δείξουμε όλες τις διαθέσιμες λειτουργίες που είναι ικανή να εκτελέσει η εφαρμογή μας. Για το σκοπό αυτό θα δημιουργήσουμε ένα υποθετικό σενάριο χρηστών και μέσα από αυτά θα προβληθούν οι λειτουργίες αλλά και η ουσιαστική ανάγκη ύπαρξής τους. Αναλυτικότερα θα υποθέσουμε ότι:

- υπάρχουν 2 συνάδελφοι (ο Χ και ο Υ) σε μία εταιρία πληροφορικής όπου καλούνται να δουλέψουν μαζί πάνω σε κάποια εργασία που τους έχει ανατεθεί.
- οι δύο αυτοί συνάδελφοι μένουν αρκετά μακριά ο ένας με τον άλλον και αποφασίζουν, λόγω πχ. Covid-19, να δουλέψουν εξ' αποστάσεως.
- ο καθένας αναλαμβάνει ένα κομμάτι και αποφασίζουν να κάνουν τακτές τηλεδιασκέψεις για να συζητάνε τα τυχόν θέματα που προκύπτουν.

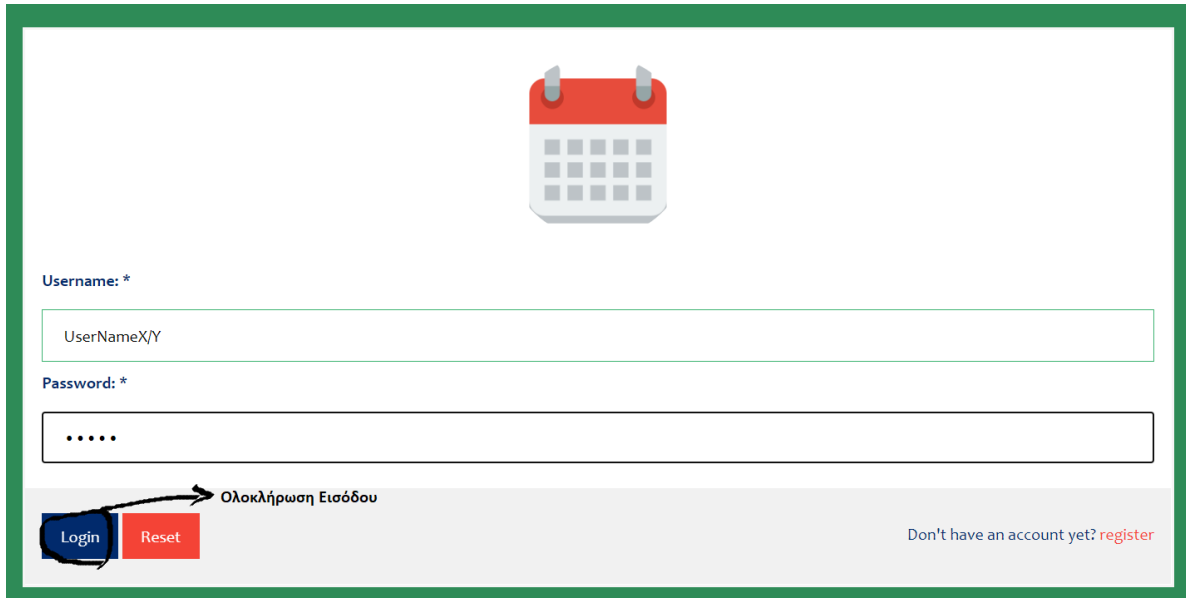
Η εφαρμογή μας λοιπόν χρησιμοποιείται και από τους δύο για να μπορέσουν να συντονίσουν τις τηλεδιασκέψεις αυτές, ώστε να ξέρουν και οι δύο πότε πρέπει να συνδεθούν.

Αρχικά θα ανοίξουν και οι δύο την εφαρμογή και θα απαιτηθεί να φτιάξουν λογαριασμούς. Όπως φαίνεται παρακάτω:



*Δημιουργία λογαριασμών του Χ και του Υ*

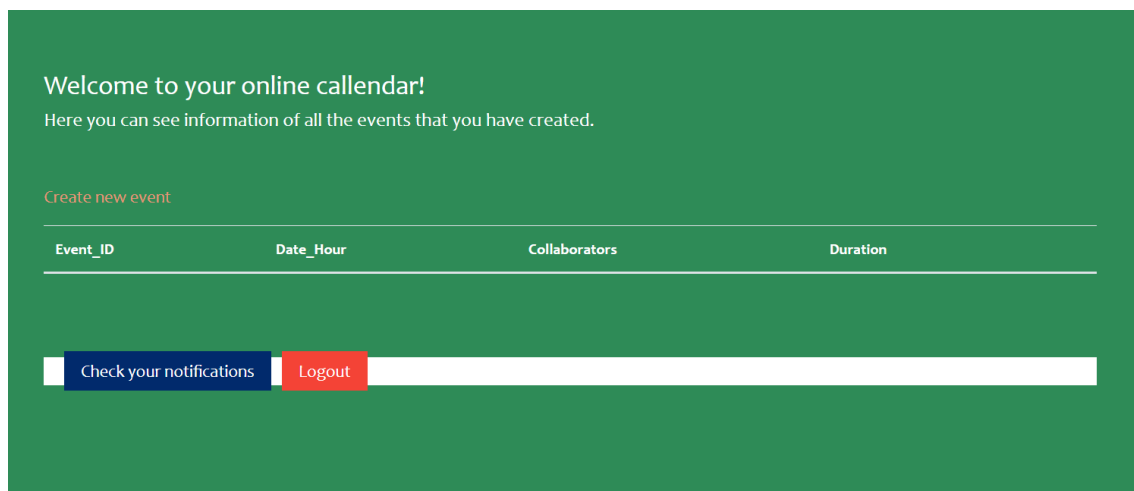
Ύστερα θα απαιτηθεί και από τους δύο να εισέλθουν στους λογαριασμούς τους για να κανονίσουν κάποια συνάντηση:



A login form with a green border. At the top center is a calendar icon. Below it, the text "Username: \*" is followed by a text input field containing "UserNameX/Y". Below that, the text "Password: \*" is followed by a password input field with five dots. At the bottom left are two buttons: a blue "Login" button and a red "Reset" button. A black arrow points from the "Login" button to the text "Ολοκλήρωση Εισόδου" (Login Completion). At the bottom right, the text "Don't have an account yet? [register](#)" is displayed.

*Σύνδεση του χρήστη στο σύστημα*

Ύστερα θα εισέλθουν στην αρχική οθόνη του ημερολογίου:




The main dashboard has a green background. At the top, it says "Welcome to your online callendar!" (note the typo) and "Here you can see information of all the events that you have created." Below this is a link "Create new event". Underneath is a table with four columns: "Event\_ID", "Date\_Hour", "Collaborators", and "Duration". At the bottom, there are two buttons: a blue "Check your notifications" button and a red "Logout" button.

Έχοντας προχωρήσει την εργασία του ο καθένας ήρθε η ώρα να προσθέσει κάποιους ραντεβού για να κάνουν την πρώτη τους τηλεδιάσκεψη. Όπως παρακάτω:  
(Έστω ότι το φτιάχνει ο Υ)



## Add a new event


Date\_Hour

10/01/2022 20:50 

Collaborators

X

Duration

1 

Create

[Go back to your events](#)

*Δημιουργία ενός ραντεβού από τον Y με συνεργάτη τον X*

Welcome to your online callendar!

Here you can see information of all the events that you have created.

[Create new event](#)

Event_ID	Date_Hour	Collaborators	Duration	
4	1/10/2022 8:50:00 PM	X	1	<a href="#">Edit Event</a>   <a href="#">Delete Event</a>

[Check your notifications](#) [Logout](#)

*Το ραντεβού που δημιουργήθηκε φαίνεται στο ημερολόγιο*

Ύστερα μία μέρα πριν από το ραντεβού ο Y αποφασίζει πως δεν μπορεί την κανονισμένη ώρα και ύστερα από συνεννόηση με τον X αποφασίζουν να την αλλάξουν:

Welcome to your online calendar!

Here you can see information of all the events that you have created.

Create new event

Event_ID	Date_Hour	Collaborators	Duration	Επεξεργασία Ραντεβού
4	1/10/2022 8:50:00 PM	X	1	<a href="#">Edit Event</a>   <a href="#">Delete Event</a>

[Check your notifications](#)

[Logout](#)

Τροποποίηση στοιχείων του ραντεβού

## Edit this Event

Date\_Hour

January 2022

Su Mo Tu We Th Fr Sa

26 27 28 29 30 31 1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31 1 2 3 4 5

Clear Today

20 50

21 51

22 52

23 53

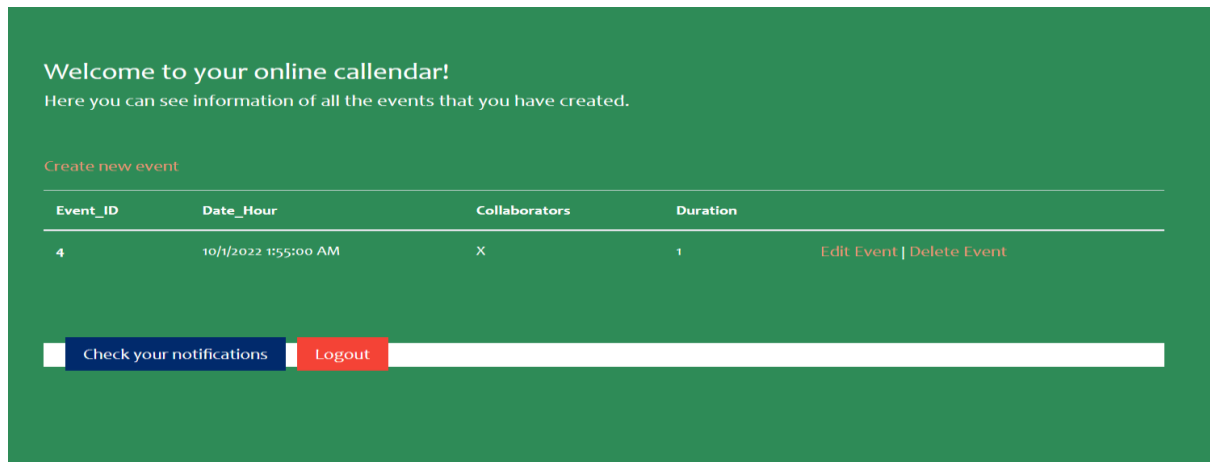
00 54

01 55

02 56

[Go back to your events](#)

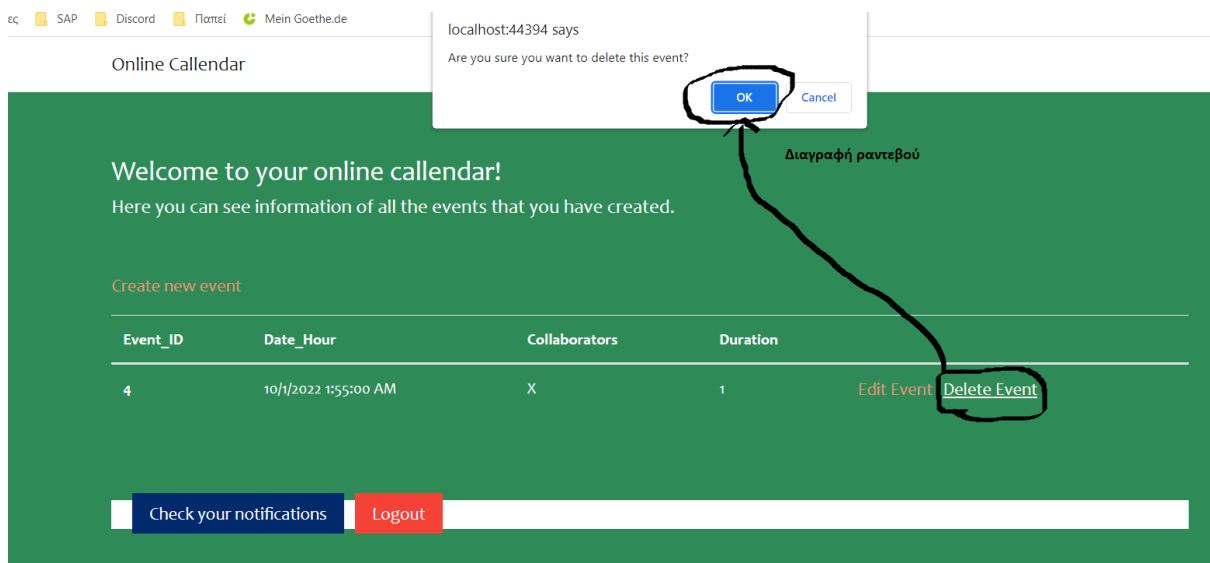
Τροποποίηση της ώρας του ραντεβού των X και Y. Μόνο ο δημιουργός Y μπορεί να αλλάζει τα στοιχεία του ραντεβού.



*Το ραντεβού των X και Y όπως αποτυπώνεται μετά την αλλαγή της ώρας του ραντεβού από τον Y*

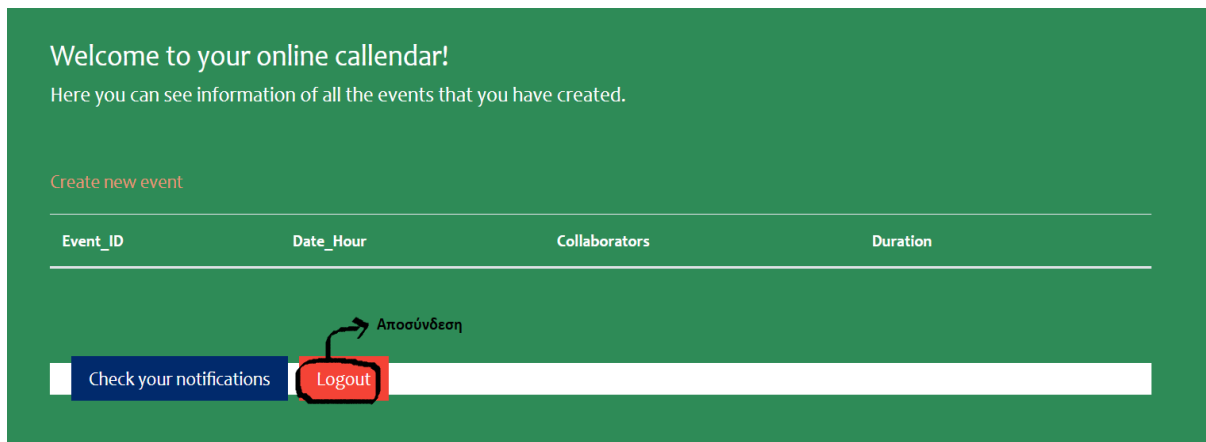
(Προφανώς μπορεί να γίνει οποιαδήποτε τέτοια αλλαγή είτε για ώρα, είτε για μέρα, είτε για μήνα, είτε για χρόνο. Επίσης μπορεί να αλλάξουν οι χρήστες που θα συμμετέχουν στο ραντεβού αλλά και η διάρκεια των ραντεβού).

Τελικά και αφού ολοκληρώσουν την τηλεδιάσκεψή τους αποφασίζουν να διαγράψουν το ραντεβού μιας και πλέον δεν εξυπηρετεί τον σκοπό του.



*Διαγραφή του ραντεβού*

Μέχρι την διευθέτηση κάποιου καινούριου ραντεβού μπορούν και οι δύο να αποσυνδεθούν από την εφαρμογή:



Παρακολουθώντας το παραπάνω σενάριο είναι εύκολο να καταλάβει κανείς τις λειτουργίες και την χρησιμότητα της εφαρμογής στην δημιουργία, επεξεργασία και διαγραφή ραντεβού στον επαγγελματικό αλλά και προσωπικό επίπεδο.

## 7. Βιβλιογραφικές Πηγές

Για το πέρας της εργασίας, χρήσιμες φάνηκαν οι ακόλουθες πηγές:

[1] [https://www.youtube.com/watch?v=t-JKbw21Gzw&ab\\_channel=ProgrammingProfessor](https://www.youtube.com/watch?v=t-JKbw21Gzw&ab_channel=ProgrammingProfessor)

[2] [https://www.youtube.com/watch?v=hZ1DASYd9rk&ab\\_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=hZ1DASYd9rk&ab_channel=freeCodeCamp.org)

[3] <https://faun.pub/asp-net-core-entity-framework-core-with-postgresql-code-first-d99b909796d7>

[4] <https://lucid.app/users/login#/loggedOut>