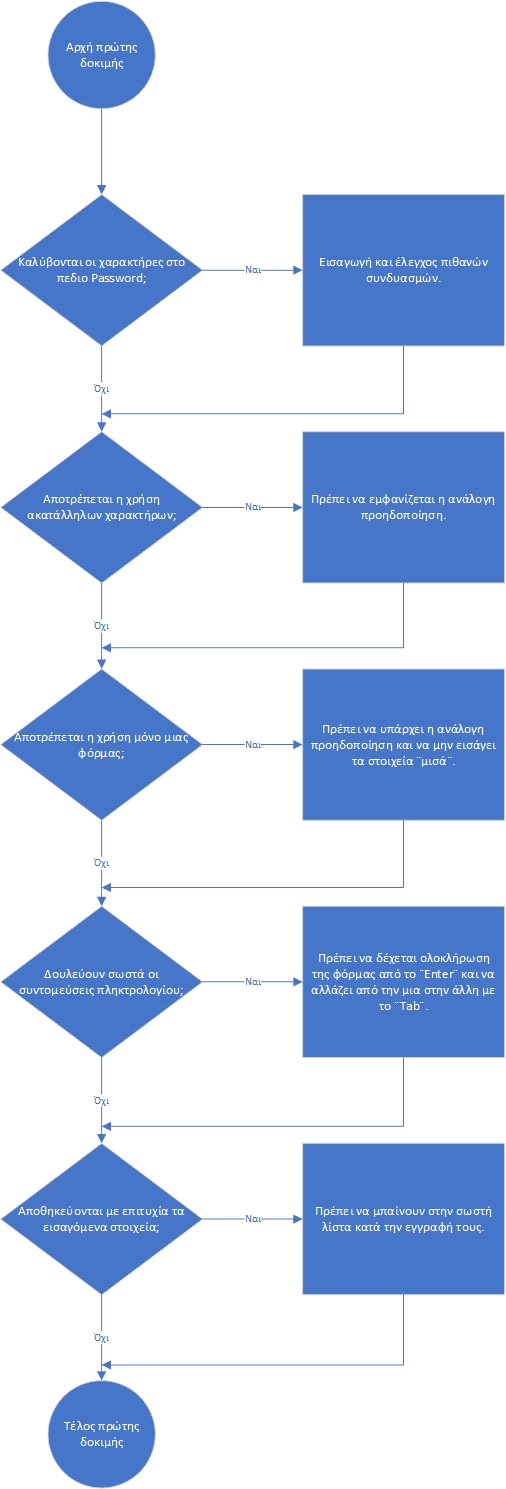
**Πρώτο μέρος της εφαρμογής:**

Το πρώτο μέρος της εργασίας που θα μου δοθεί για έλεγχο λειτουργικότητας και ανίχνευση προβλημάτων θα είναι το log-in screen (οθόνη εισόδου χρήστη) και έχει ονομαστεί από τους προγραμματιστές ως ¨SSRI¨.

Σε αυτή την οθόνη ο χρήστης θα εισάγει τα στοιχεία του για να μπορεί να αποκτήσει πρόσβαση στην εφαρμογή μέσα από μία φόρμα η οποία θα του ζητάει τα εξής στοιχεία: Username και Password, όπου username θα είναι η φόρμα στην οποία θα εισάγεται το όνομα χρήστη και password η φόρμα η οποία θα συμπληρώνεται με τον κωδικό αυτού του χρήστη.

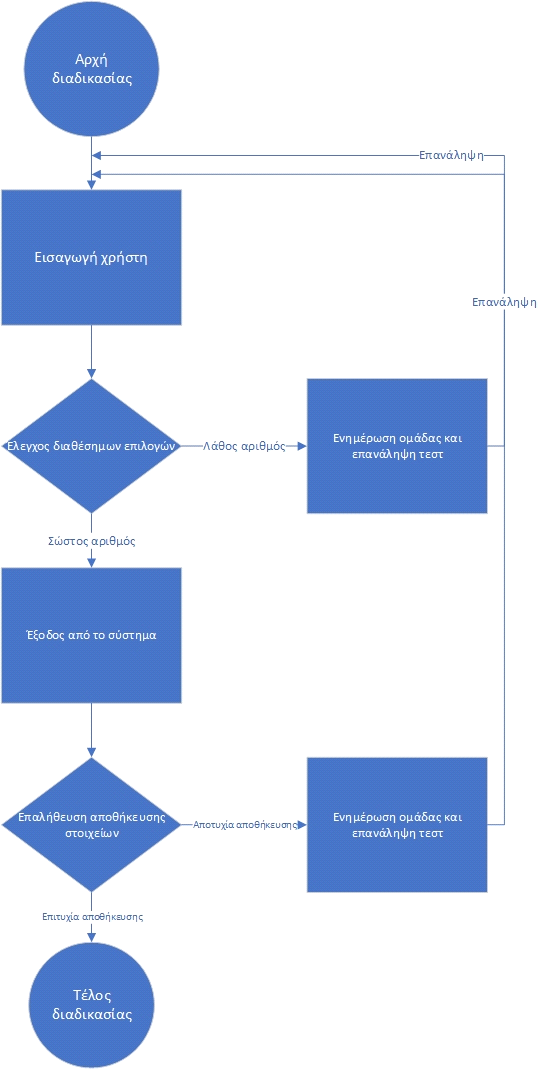
Οι περιπτώσεις τεστ θα ακολουθήσουν ως εξής:

* Έλεγχος της φόρμας password εάν τηρεί τις βασικές αρχές προστασίας του χρήστη κρύβοντας τον κωδικό με κάποιον τρόπο.
* Τι χαρακτήρες θεωρούνται αποδεκτοί από το σύστημα; Σε περίπτωση ακατάλληλου input από τον χρήστη υπάρχει τρόπος να τον ενημερώσει η να τον προτρέψει;
* Έλεγχος της οθόνης όσο αναφορά άδεια πεδία. Ο χρήστης πρέπει να προτρέπεται από το να χρησιμοποιεί μόνο μία φόρμα.
* Έλεγχος λειτουργικότητας κουμπιών και λειτουργίες πληκτρολογίου όπως η αλλαγή φόρμας με το πλήκτρο tab και αποδοχή εισόδου μέσω του enter.
* Έλεγχος αποθήκευσης των στοιχείων που εισάγονται από τον χρήστη, έτσι ώστε να καταγράφονται από το σύστημα.
* Έλεγχος αποθήκευσης των προαναφερόμενων στοιχείων στις σωστές λίστες. (Τα ονόματα στο username, τους κωδικούς στα password κ.λπ.)
* Έλεγχος ευχρηστίας. (Π.χ. εάν το παράθυρο είναι αρκετά ορατό, κατάλληλου μεγέθους και οι φόρμες δεν πέφτουν πολύ κοντά μεταξύ τους μπερδεύοντας τον χρήστη ή κάνοντας την εμπειρία δυσάρεστη).

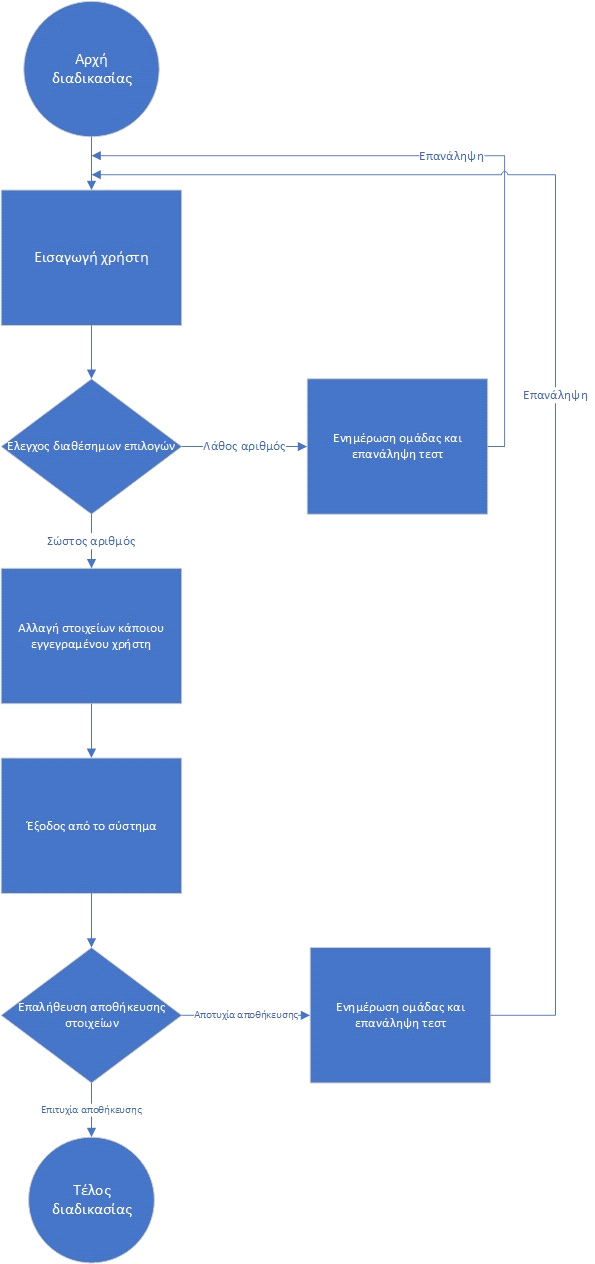
**Πρώτο διάγραμμα γενικού πλάνου τεστ (Log-in screen):**

Κάθε ένας χρήστης του συστήματος θα έχει τα δικά του επίπεδα πρόσβασης ανάλογα τον ρόλο του. Για να γίνει σίγουρο πως αυτό δουλεύει θα εισάγω δύο χρήστες στο σύστημα μέσω της βάσης δεδομένων, θα τους δοθεί διαφορετικός ρόλος στον καθένα και θα ελεγχθούνε αν οι δυνατότητες που δίνονται στους χρήστες αναλογούν στον τίτλο τους. Εάν το πρόγραμμα εμφανίσει τις σωστές οθόνες ,ανάλογα με τον εισαγόμενο χρήστη, αποθηκεύοντας αλλαγές και προσθήκες, η αποτρέποντάς τον ανάλογα, τότε το τεστ θα θεωρείται επιτυχία.

**Διάγραμμα τεστ εισαγωγής απλού χρήστη:**

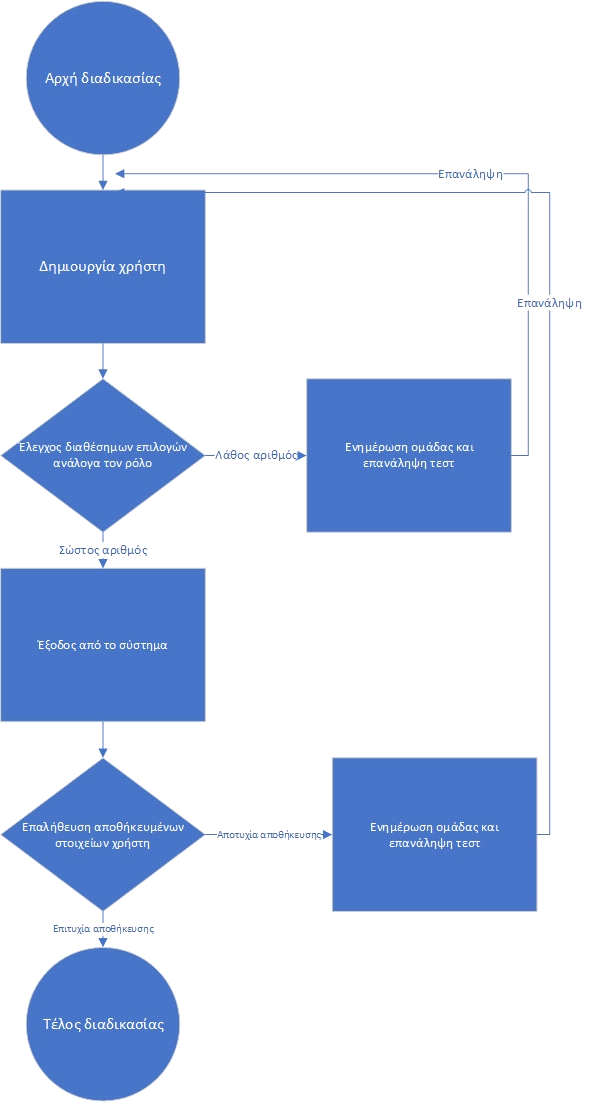


**Διάγραμμα τεστ** **εισαγωγής admin:**

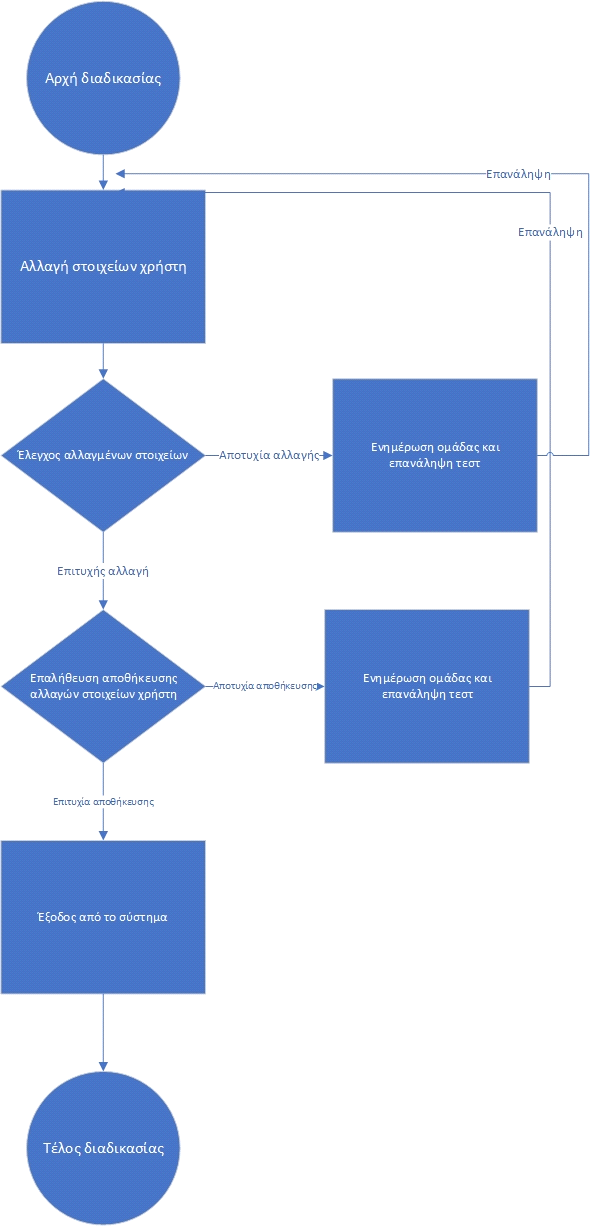


Ο διαχειριστής, καθώς και το ίδιο το πρόγραμμα, θα πρέπει να έχει την δυνατότητα απαραίτητων αλλαγών όπως τον εμπλουτισμό στοιχείων, πρόσθεση και διαγραφή αυτών από το σύστημα, καθώς και την ορθή εκτέλεση αυτών. Για παράδειγμα, οι διαγραφές δεν θα πρέπει να αφήνουν κενά μνήμης καθώς, οι προσθήκες θα πρέπει να γίνονται μέσα στα όρια του χρόνου απόκρισης κ.λπ. Το τεστ θα γίνει με τις εισαγωγές που αναφέρονται προηγουμένως. Εάν το πρόγραμμα προσθέσει με επιτυχία τα εισαγόμενα δεδομένα, τα αλλάξει και τέλος τα διαγράφει χωρίς υπολείμματα στοιχείων στην μνήμη, μέσα στα όρια απόκρισης που ορίστηκε και επιστραφεί η οθόνη, με όλες τις αλλαγές οι ποίες έγιναν σε κάθε μια από αυτές τις περιπτώσεις, τότε το τεστ θεωρείται επιτυχία.

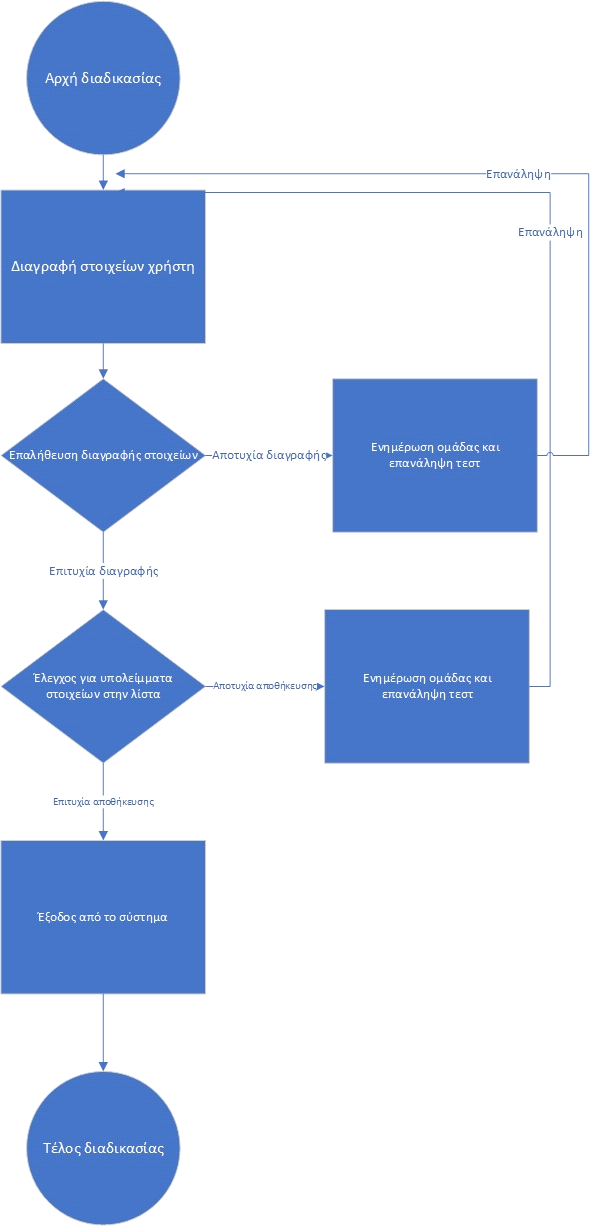
**Διάγραμμα τεστ** **δημιουργίας ενώς χρήστη:**



**Διάγραμμα τεστ** **αλλαγής στοιχείων χρήστη:**



**Διάγραμμα τεστ** **διαγραφής στοιχείων χρήστη:**



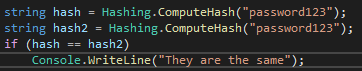
**Περιγραφή ανάλυση και αποτελέσματα Unit Testing του Log-In Screen**

Σε αυτό το μέρος του εγγράφου γίνεται η ανάλυση των test που πραγματοποιήθηκαν προκειμένου να εξασφαλιστεί η σωστή λειτουργία της εφαρμογής. Σε αυτή περιέχονται οι μέθοδοι, ο αριθμός επαναλήψεων και τέλος είσοδοι και έξοδοι αυτών οι οποίοι έπαιξαν καθοριστικό ρόλο στην ημερομηνία κυκλοφορίας της πρώτης έκδοσης της εφαρμογής.

Η μέθοδος που χρησιμοποιήθηκε εμπεριείχε τα εξής test:

* Έλεγχος hash του κωδικού χρήστη.
* Username generator το οποίο καθορίζει το Username ανάλογα με την είσοδο του χρήστη.
* Συνθήκες bool που ελέγχουν την εγκυρότητα των εισόδων του χρήστη.
* Τον τρόπο με τον οποίο ελέγχθηκαν οι bool είσοδοι.

Ξεκινώντας με τον έλεγχο του hash οι είσοδοι δεν ήταν σημαντικοί ως προς το test όσο ήτανε το αποτέλεσμα που προέκυπτε. Προκειμένου αυτό το test να διεξαχθεί με επιτυχία οι όροι είχανε ως εξής: αφού ο tester εισήγαγε τον ίδιο κωδικό και στο hash και στο hash2, όταν έτρεχε ο κώδικας έλεγχε εάν το υπολογισμένο hash είναι ίδιο με το άλλο. Αφότου λοιπόν πραγματοποιούταν αυτός ο έλεγχος, εάν όλα πήγαιναν καλά, η έξοδος του προγράμματος έπρεπε να είναι ¨They are the same¨ επιβεβαιώνοντας ότι όντως το hash είναι ίδιο και ότι το test πραγματοποιήθηκε με επιτυχία. Ακολουθεί screenshot του κώδικα:



Οι είσοδοι που ακολούθησαν ήταν: ¨password123¨,¨kodikos345¨,¨simplepass¨. Οι τρείς κωδικοί χρησιμοποιήθηκαν από δύο φορές ο καθένας, συνολικά 6 test, και υπήρξε 100% επιτυχία.

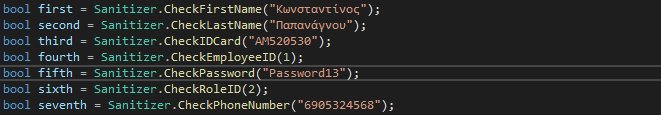
Συνεχίζοντας, το επόμενο test πραγματοποιήθηκε επάνω στο username generator όπου ο γίνεται είσοδος κάποιου ονόματος και επιθέτου, ως αποτέλεσμα λαμβάνεται ένα τυχαία δομημένο Username το οποίο αποτελείται από δύο (2) γράμματα του ονόματος και τέσσερα (4) γράμματα του επιθέτου. Οι είσοδοι και έξοδοι κάτω του screenshot αποτελούνε παραστατικό παράδειγμα. Ακολουθεί screenshot του κώδικα:

username_gen_nakanwparadigmata

|  |
| --- |
| Είσοδοι:   * (ΠΑΝΑΓΙΩΤΗΣ,ΣΚΛΙΔΑΣ) * (ΑΘΑΝΑΣΙΟΣ,ΜΕΛΙΣΣΟΣ) * (ΚΩΝΣΤΑΝΤΙΝΟΣ,ΠΑΠΑΝΑΓΝΟΥ)   Έξοδοι:   * paskli,paklid * a8meli,a8mell * kwpapa,kwpana |

Το test έγινε έξι (6) φορές και είχε απόλυτη επιτυχία. Οι μόνες παρατηρήσεις που έγιναν ήταν ότι λόγο σύνδεσης ελληνικού αλφαβήτου με αγγλικούς χαρακτήρες πολλές φορές το τυχαία παραγόμενο username είναι δυσνόητο από τον χρήστη ιδιαίτερα όταν υπάρχουν ονόματα που συμπεριλαμβάνουν τα γράμματα ¨ξ¨ και ¨θ¨ καθώς τα ανάλογα στο αγγλικό αλφάβητο αντικαταστήθηκαν με το ¨3¨,¨8¨ ανάλογα.

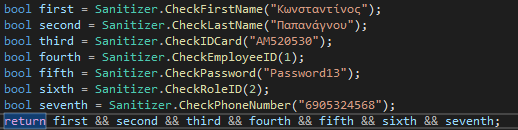
Κλείνοντας το πρώτο test στα τελευταία μέρη του κώδικα έγινε ένας γενικός έλεγχος όλων των πεδίων, αυτά είναι: Όνομα, Επίθετο, Id χρήστη, Επίπεδο πρόσβασης Κωδικός χρήστη, Ρόλος του χρήστη, Τηλέφωνο χρήστη. Με αυτόν τον έλεγχο τα πεδία προς συμπλήρωση ελέγχθηκαν κατάλληλα με διάφορους εισόδους για να εξασφαλισθεί η αξιοπιστία αυτών, έτσι ώστε να αποφευχθεί οποιαδήποτε λάθος χρήση ή είσοδος. Ακολουθεί screenshot του κώδικα:



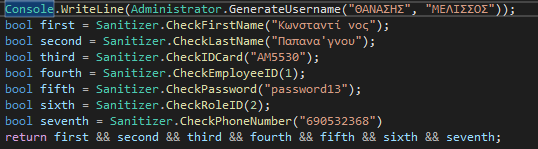
Οι έλεγχοι που ακολούθησαν ήτανε σε μορφή εισόδων και έχουν ως εξής: άδεια ή ελλείπει στοιχεία, λάθος στοιχεία και τυχαία κενά ανάμεσα στα στοιχεία. Ο έλεγχος έγινε με την λογική ανίχνευσης των λαθών για την ομαλή λειτουργία της εφαρμογής σε περίπτωση λάθους εισόδου. Ο έλεγχος έγινε με βάση breakpoints όπου το κάθε ένα από αυτά τα στοιχεία εάν θεωρούτανε αποδεκτά από το πρόγραμμα επέστρεφαν true αλλιώς το πρόγραμμα ενημέρωνε κατάλληλα επιστρέφοντας false. Ο έλεγχος έγινε με την βοήθεια της εξής γραμμής κώδικα:

apotelesmata_EKSIGISI

Σε αυτή, ελέγχονται όλες οι bool καταστάσεις πολύ πιο άμεσα εξυπηρετώντας ως ένα γρήγορο και αξιόπιστο εργαλείο ελέγχου, όπου εάν υπήρχε έστω και ένα λάθος, το breakpoint της γραμμής θα ενημέρωνε με true η false κατά την διάρκεια του debugging προειδοποιώντας ανάλογα. Ακολουθεί screenshot ολόκληρου του κώδικα με μέθοδο boole:



Παραδείγματα από tests στα οποία το πρόγραμμα έπρεπε να ενημερώσει τον ελεγκτή ότι υπάρχει λάθος και χρησιμοποιήθηκαν κατά την διάρκεια των test:



Εδώ το πρόγραμμα έχει ως έξοδο false καθώς υπάρχουνε προβληματικές είσοδοι σε όλα τα πεδία κειμένου, πιο αναλυτικά υπάρχει κενό στο όνομα του χρήστη, λάθος χαρακτήρας το επίθετο, ελλιπείς id κάρτας, αποκλειστική χρήση μη κεφαλαίων στον κωδικό και αριθμό τηλεφώνου που αποτελείται από 9 χαρακτήρες. Επομένως κάνοντας αναδρομή και χρησιμοποιώντας breakpoints για debugging μπορούμε να δούμε ότι:

|  |  |
| --- | --- |
| first | false |
| second | false |
| third | false |
| fourth | true |
| fifth | false |
| sixth | true |
| seventh | false |

Στην περίπτωση hash όταν γίνεται έλεγχος και ο κωδικός διαφέρει του άλλου αλλάζουν τα hash οπότε δεν καλείται ποτέ η γραμμή ¨They are the same¨ καθώς το hash ==hash2 μετατρέπεται σε false. Ακολουθεί παράδειγμα άνισης ισοδυναμίας:

