

Επιλέξαμε την διεξαγωγή των Test Cases μέσω της black box μεθόδου. Το Black Box Testing είναι, επίσης, γνωστό σαν συμπεριφορική μέθοδος, eye-to-eye testing, μέθοδος του αδιαφανούς κουτιού ή μέθοδος του κλειστού κουτιού, με βάση τις προδιαγραφές.

Είναι μια μέθοδος Ελέγχου Λογισμικού που αναλύει τη λειτουργικότητα ενός λογισμικού/εφαρμογής, χωρίς να γνωρίζει πολλά για την εσωτερική δομή/σχεδιασμό του στοιχείου που δοκιμάζεται και συγκρίνει την τιμή εισόδου με την τιμή εξόδου. Η κύρια εστίαση του Black Box Testing είναι στην λειτουργικότητα του συστήματος στο σύνολό του.

Τα βασικά πλεονεκτήματα της μεθόδου που μας οδήγησε στην επιλογή της είναι τα εξής:

- Ο tester και ο developer είναι ανεξάρτητοι ο ένας από τον άλλο και η δοκιμή είναι ισορροπημένη και χωρίς προκαταλήψεις.
- Ο tester δεν χρειάζεται να έχει λεπτομερή τεχνική κατάρτιση του Συστήματος.
- Οι έλεγχοι γίνονται από την πλευρά του τελικού χρήστη, επειδή μας ενδιαφέρει η κάλυψη των αναγκών του.
- Ο έλεγχος βοηθά στον εντοπισμό αοριστίας και αντιφάσεων στις λειτουργικές προδιαγραφές.
- Τα Test Cases μπορούν να σχεδιαστούν από την στιγμή που θα ολοκληρωθούν οι λειτουργικές προδιαγραφές.

Ως εκ τούτου, καταλήξαμε στην δημιουργία ενός πίνακα με στήλες: Test Case ID, Test Case Περιγραφή, Περιπτώσεις Ελέγχου, Θετική/Αρνητική Ροή, Προϋπόθεση, Βήματα Testing, Δεδομένα, Αναμενόμενο Αποτέλεσμα, Status και Πραγματικό Αποτέλεσμα.

**Test Case ID:** Μοναδικό ID για κάθε Test Case. Αρχίζει με το ακρωνύμιο TC, τον τίτλο χωρισμένο με κάτω παύλες και τον αριθμό βάσει της σειράς ελέγχου. Κάθε σύνολο Test Case ξεκινάει από το 01.

**Test Case Περιγραφή:** Σύντομη περιγραφή των βημάτων του Test Case και του αναμενόμενου αποτελέσματος.

**Περιπτώσεις Ελέγχου:** Αφορά Test Cases που έχουν περιορισμούς και παίρνει σαν όρισμα: Άνω Όριο, Κάτω Όριο, Έγκυρη Τιμή, Άκυρη Τιμή. Π.χ.: αν μια ομάδα μπορεί να απαρτίζεται από 2 έως 10 άτομα, έχει ως Άνω Όριο το 10, Κάτω Όριο το 2, Έγκυρη Τιμή το 5 και Άκυρη Τιμή το 11.

**Θετική/Αρνητική Ροή:** Η συγκεκριμένη στήλη παίρνει ως ορίσματα Θετική/Αρνητική. Αναφερόμαστε στο τι θέλει να δει ο χρήστης και όχι ο developer στο αναμενόμενο αποτέλεσμα. Π.χ. στο TC\_Login\_NN, η πληκτρολόγηση λανθασμένου κωδικού έχει αρνητικό αντίκτυπο για τον χρήστη, καθώς, δεν καταλήγει σε επιτυχή είσοδό του. Στο TC\_Login\_NM, η πληκτρολόγηση σωστού κωδικού έχει θετικό αντίκτυπο για τον χρήστη, καθώς, καταλήγει σε επιτυχή είσοδό του. Ουσιαστικά, μέσω της Αρνητικής Ροής προσπαθούμε να προβλέψουμε μη αποδεκτή σειρά από τον χρήστη (λανθασμένος κωδικός/ username/ email, κενά πεδία, άκυρος αριθμός κ.ο.κ.)

**Προϋπόθεση:** Η κατάσταση της εφαρμογής που ελέγχεται και απαρίθμηση των παραγόντων που απαιτούνται για να προχωρήσουμε στα επόμενα βήματα του ελέγχου.

**Βήματα Testing:** Αριθμημένες αναλυτικές οδηγίες για τον τρόπο διεξαγωγής του ελέγχου.

**Δεδομένα:** Τα δεδομένα που τροφοδοτούν την εφαρμογή, με σκοπό το αναμενόμενο αποτέλεσμα. Σε αυτά υπάγονται, τα δεδομένα που πληκτρολογεί ο χρήστης κατά την προσθήκη διεύθυνσης (TC\_AddNewAddress\_NN), τα API's Τράπεζας (TC\_Payment\_NN) και χαρτών (TC\_AddNewAddress\_NM) κ.ο.κ.

**Αναμενόμενο Αποτέλεσμα:** Τι αναμένουμε να δούμε από το συγκεκριμένο Test Case.

**Status:** Η κατάσταση του Test Case. Λαμβάνει συγκεκριμένα ορίσματα:  
Pass: Το Πραγματικό Αποτέλεσμα συνάδει με το Αναμενόμενο.  
Fail: Υπάρχει Απόκλιση μεταξύ του Πραγματικού και του Αναμενόμενου Αποτελέσματος.  
Incomplete: Για κάποιο λόγο, δεν έχει ολοκληρωθεί το testing.  
Blocked: Το testing είναι μπλοκαρισμένο, λόγω προϋπάρχοντος ανοιχτού ζητήματος.  
No Run: Το testing δεν έχει ξεκινήσει.

**Πραγματικό Αποτέλεσμα:** Τι βλέπουμε, εν τέλει, από το συγκεκριμένο Test Case, κατά τον τελευταία εκτέλεσή του.