**Robustness-diagram-v0.1**

BLOCKO

**Μέλη Ομάδας:**

**Όνομα**: Κωνσταντίνος

**Επώνυμο**: Καρακίτσος

**ΑΜ**: 1093381

**Όνομα**: Βασίλειος

**Επώνυμο**: Μιχανετζής

**ΑΜ:** 1093433

**Όνομα**: Νυμφοδώρα

**Επώνυμο:** Μπουζιάνη

**ΑΜ**: 1097429

**Όνομα**: Μιχαήλ

**Επώνυμο:** Σοκολάκης

**ΑΜ**: 1072589

**Όνομα**: Σωτήριος

**Επώνυμο**: Χασιώτης

**ΑΜ**: 1093510

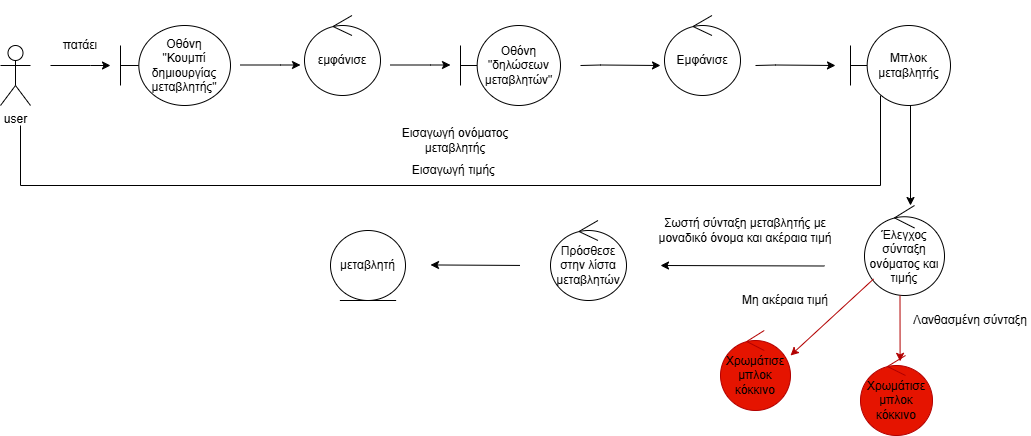
**Εργαλεία :**

Τα εργαλεία που χρησιμοποιήσαμε για την σύνταξη των τεχνικών κειμένων είναι το Microsoft Word και για την δημιουργία των διαγραμμάτων το draw.io.

**Editor:** Νυμφοδώρα Μπουζιάνη, Σωτήριος Χασιώτης

**Contributors:** Κωσταντίνος Καρακίτσος, Νυμφοδώρα Μπουζιάνη, Μιχαήλ Σοκολάκης

**Peer Reviewer:** Βασίλειος Μιχανετζής



**Σχ1. Διάγραμμα ευρωστίας για Use Case 1**

**Βασική Ροή: Ο χρήστης δημιουργεί μια νέα μεταβλητή .**

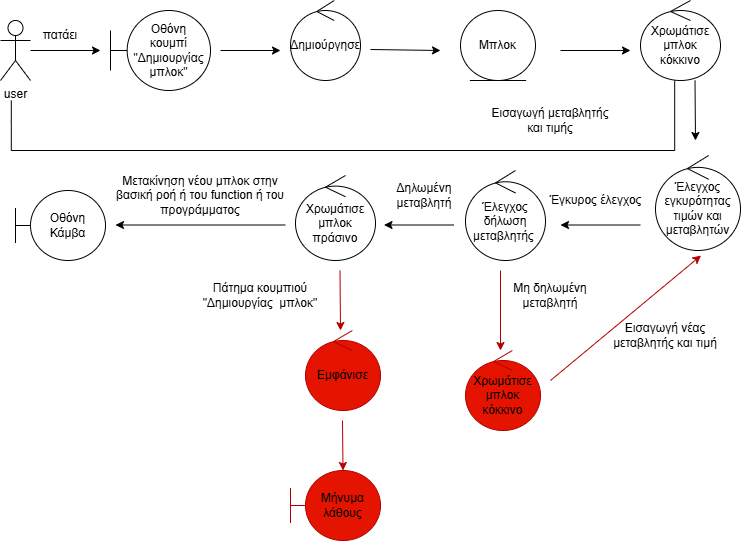
Ο χρήστης επιθυμεί να δημιουργήσει μια νέα μεταβλητή και πατάει το κουμπί δημιουργίας μεταβλητής. Εμφανίζεται το μπλοκ της μεταβλητής στον χώρο των δηλώσεων μεταβλητών και ο χρήστης καλείται να εισάγει το όνομα της μεταβλητής και την τιμή της . Αφού γίνει ο έλεγχος για την εγκυρότητα της σύνταξης του ονόματος και της τιμής της μεταβλητής τότε το σύστημα προσθέτει τη μεταβλητή στη λίστα μεταβλητών και η διαδικασία της δημιουργίας μεταβλητής ολοκληρώνεται επιτυχώς.

**Εναλλακτική Ροή 1: Ο χρήστης εισάγει όνομα μεταβλητής με λάθος σύνταξη.**

Ο χρήστης εισάγει μη έγκυρο όνομα μεταβλητής και το σύστημα χρωματίζει το μπλοκ κόκκινο.

**Εναλλακτική Ροή 2: Ο χρήστης εισάγει μη ακέραια τιμή στην μεταβλητή.**

Ο χρήστης εισάγει μην ακέραια τιμή στην μεταβλητή και το σύστημα χρωματίζει το μπλοκ κόκκινο.



**Σχ2. Διάγραμμα ευρωστίας για Use Case 2**

**Βασική Ροή: Ο χρήστης θέλει να δημιουργήσει ένα νέο μπλοκ.**

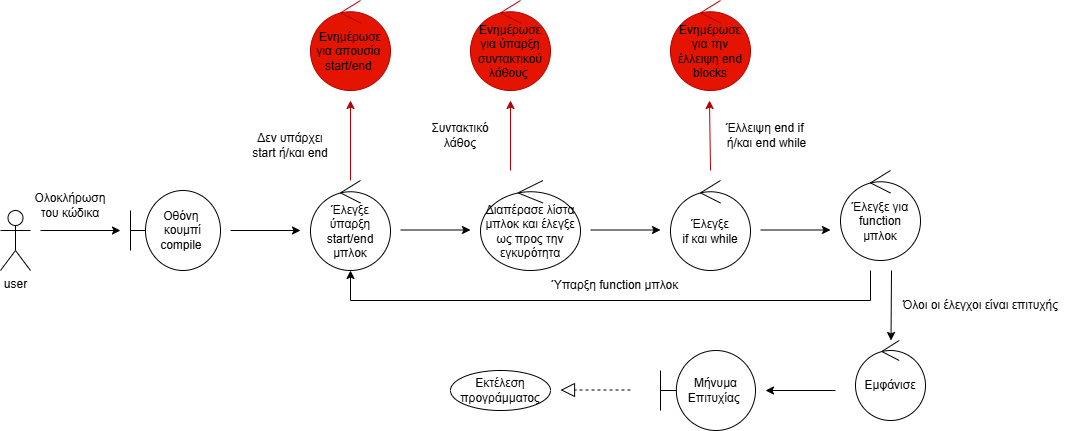
Ο χρήστης πατάει το κουμπί Δημιουργίας μπλοκ και δημιουργείται καινούργιο μπλοκ το οποίο χρωματίζεται κόκκινο. Ο χρήστης καλείται να εισάγει την επιθυμητή μεταβλητή και τιμή και εφόσον αυτά που εισήγαγε είναι έγκυρα και η μεταβλητή είναι δηλωμένη τότε το σύστημα χρωματίζει το μπλοκ πράσινο ως ένδειξη εγκυρότητας. Τέλος ο χρήστης πρέπει να μετακινήσει το μπλοκ που δημιούργησε στην βασική ροή είτε του προγράμματος του είτε του function του.

**Εναλλακτική Ροή 1: Ο χρήστης δεν έχει δηλώσει την μεταβλητή.**

Ο έλεγχος για την δηλωμένη μεταβλητή είναι ανεπιτυχής και το μπλοκ εξακολουθεί να είναι κόκκινο οπότε ο χρήστης καλείται να εισάγει εκ νέου άλλη μεταβλητή.

**Εναλλακτική Ροή 2: Ο χρήστης θέλει να δημιουργήσει ένα νέο μπλοκ ενώ δεν έχει μετακινήσει το προηγούμενο μπλοκ που δημιούργησε από τον χώρο δημιουργίας.**

Ο χρήστης ενημερώνεται μέσω μηνύματος σφάλματος ότι πρέπει να μετακινήσει πρώτα το προηγούμενο μπλοκ που δημιούργησε από τον χώρο δημιουργίας και έπειτα να δημιουργήσει καινούργιο μπλοκ.



**Σχ3. Διάγραμμα ευρωστίας για Use Case 3**

**Βασική Ροή: Ο χρήστης θέλει να κάνει compile τον κώδικα του.**

Ο χρήστης πατάει το κουμπί compile και το σύστημα αφού ελέγξει για την ύπαρξη end και start μπλοκ διαπερνά την λίστα μπλοκ ελέγχοντας την εγκυρότητα της σύνταξης των μπλοκ και συνεχίζει με τον έλεγχο ορθότητας των δομών if και των βρόγχων while. Έπειτα ελέγχει για την ύπαρξη function μπλοκ και εφόσον υπάρχει διαπερνά την λίστα μπλοκ της υπορουτίνας και επαναλαμβάνονται οι προηγούμενοι έλεγχοι από την αρχή. Οι έλεγχοι εφόσον είναι επιτυχής εμφανίζεται μήνυμα επιτυχίας.

**Εναλλακτική Ροή 1: Ο χρήστης δεν έχει τοποθετήσει στο πρόγραμμα του start ή/και end μπλοκ.**

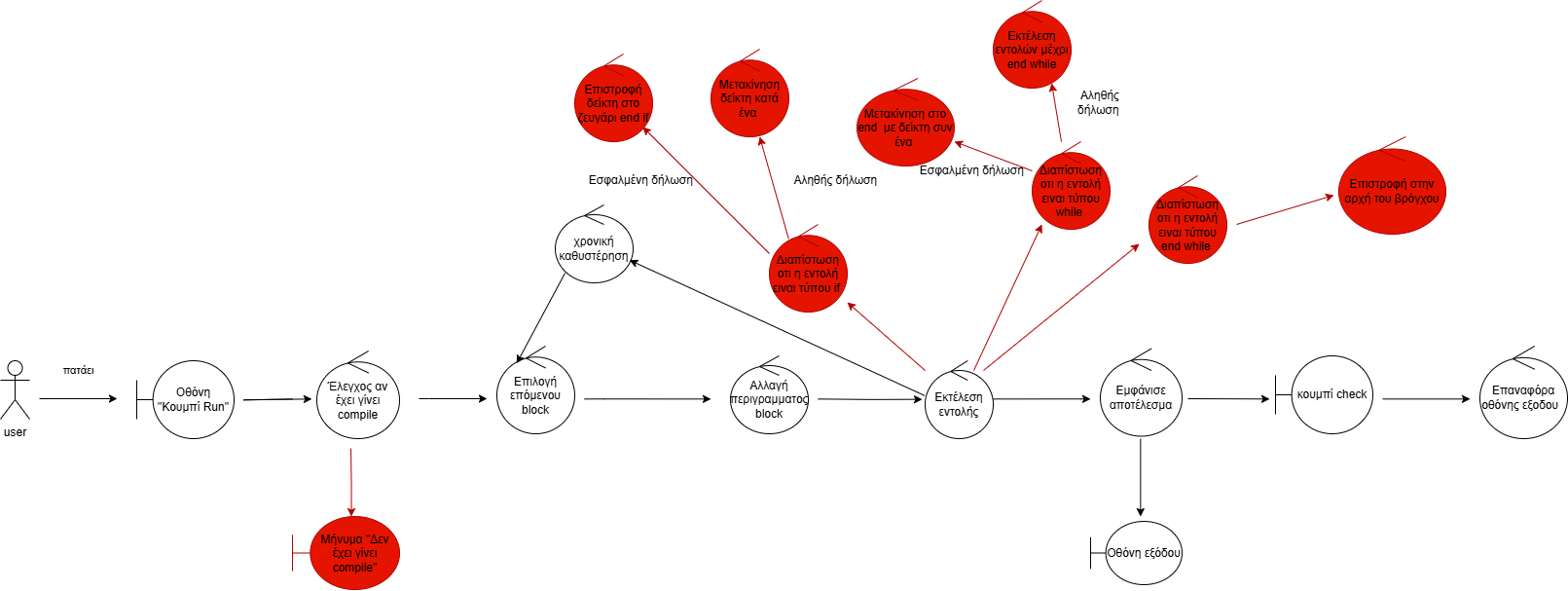
Το σύστημα αφού διαπιστώνει ότι δεν υπάρχει start ή/και end μπλοκ και ενημερώνει τον χρήστη για την απουσία τους.

**Εναλλακτική Ροή 2: Ο χρήστης έχει κάνει συντακτικό λάθος.**

Το σύστημα διαπερνά την λίστα των μπλοκ και βρίσκει συντακτικό λάθος με αποτέλεσμα να ενημερώνει τον χρήστη για την ύπαρξη των συντακτικών σφαλμάτων του .

**Εναλλακτική Ροή 3: Ο χρήστης δεν έχει τοποθετήσει στο πρόγραμμα του end if ή/και end while μπλοκ.**

Το σύστημα εφόσον διαπιστώνει ότι δεν υπάρχει end if ή/και end while μπλοκ, ενημερώνει τον χρήστη συγκεκριμένα σε ποια σημεία εντοπίζεται η απουσία τους.

****

**Σχ4. Διάγραμμα ευρωστίας για Use Case 4**

**Βασική Ροή: Ο χρήστης εκτελεί το πρόγραμμα του.**

Ο χρήστης πατάει το κουμπί run και γίνεται έλεγχος για το αν έχει γίνει πρώτα compile. Στην συνέχεια ξεκινάει η εκτέλεση του προγράμματος ενώ εμφανίζεται φωτεινό περίγραμμα γύρω από την εντολή προς εκτέλεση. Με μια μικρή καθυστέρηση το σύστημα εκτελεί το κάθε μπλοκ εντολής οπού θα εμφανίζονται οι τιμές των μεταβλητών στην οθόνη εξόδου και μετακινεί τον δείκτη στην επόμενη εντολή. Όταν φτάσουμε στο end μπλοκ, ο χρήστης βλέπει στην οθόνη εξόδου τις τελευταίες τιμές των μεταβλητών και το σύστημα ενεργοποιεί το κουμπί εκκαθάρισης .

**Εναλλακτική Ροή 1: Ο χρήστης δεν έχει κάνει compile και εκτελεί το πρόγραμμα του.**

Το σύστημα διαπιστώνει ότι δεν έχει γίνει compile και εμφανίζει στον χρήστη μήνυμα σφάλματος.

**Εναλλακτική Ροή 2: Ο χρήστης έχει χρησιμοποιήσει στο πρόγραμμα του δομή ελέγχου if και η συνθήκη if αποτελεί αληθής δήλωση.**

Το σύστημα διαπιστώνει ότι υπάρχει εντολή τύπου if με αληθής συνθήκη και προχωράει κατά ένα ο δείκτης.

**Εναλλακτική Ροή 2.1 (Εναλλακτική ροή της εναλλακτικής ροής 2): Ο χρήστης έχει χρησιμοποιήσει στο πρόγραμμα του δομή ελέγχου if αλλά η συνθήκη if είναι εσφαλμένη.**

Το σύστημα διαπιστώνει ότι υπάρχει εντολή τύπου if με εσφαλμένη δήλωση και επιστρέφει τον δείκτη στο ζευγάρι end if.

**Εναλλακτική Ροή 3: Ο χρήστης έχει χρησιμοποιήσει στο πρόγραμμα του** **βρόγχο while και αποτελεί σωστή δήλωση.**

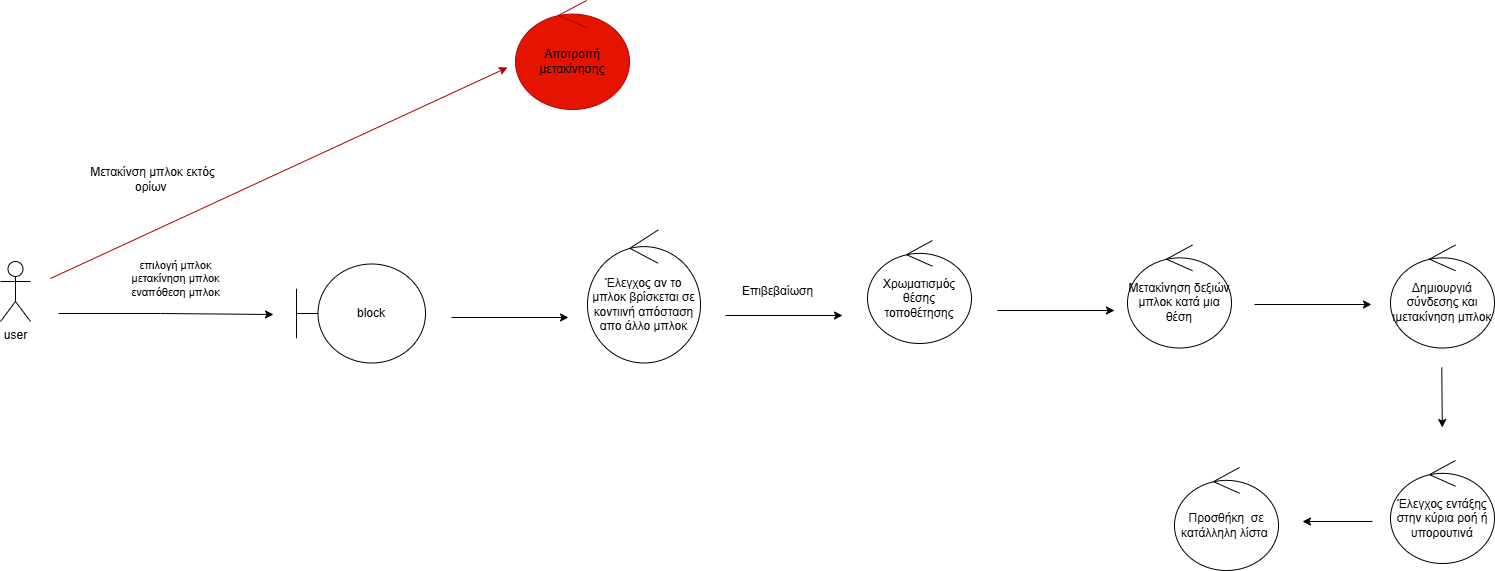
Το σύστημα διαπιστώνει ότι υπάρχει εντολή τύπου while και εκτελούνται όλες οι εντολές μέχρι και end while.

**Εναλλακτική Ροή 3.1 (Εναλλακτική ροή της εναλλακτικής ροής 3):** **Ο χρήστης έχει χρησιμοποιήσει στο πρόγραμμα του** **βρόγχο while και αποτελεί εσφαλμένη δήλωση.**

Το σύστημα διαπιστώνει ότι υπάρχει εντολή τύπου while, εκτελούνται όλες οι εντολές μέχρι και end while και αφού το σύστημα διαπιστώσει ότι η συνθήκη while έχει δηλωθεί λανθασμένα υπάρχει μετακίνηση του δείκτη στο end με αύξηση κατά ένα.

**Εναλλακτική Ροή 4: Ο χρήστης έχει χρησιμοποιήσει στο πρόγραμμα του** **βρόγχο while και το πρόγραμμα έχει φτάσει σε τύπο μπλοκ end while.**

Το σύστημα διαπιστώνει ότι η εντολή είναι τύπου end while και επιστρέφει τον δείκτη στην αρχή του βρόγχου.



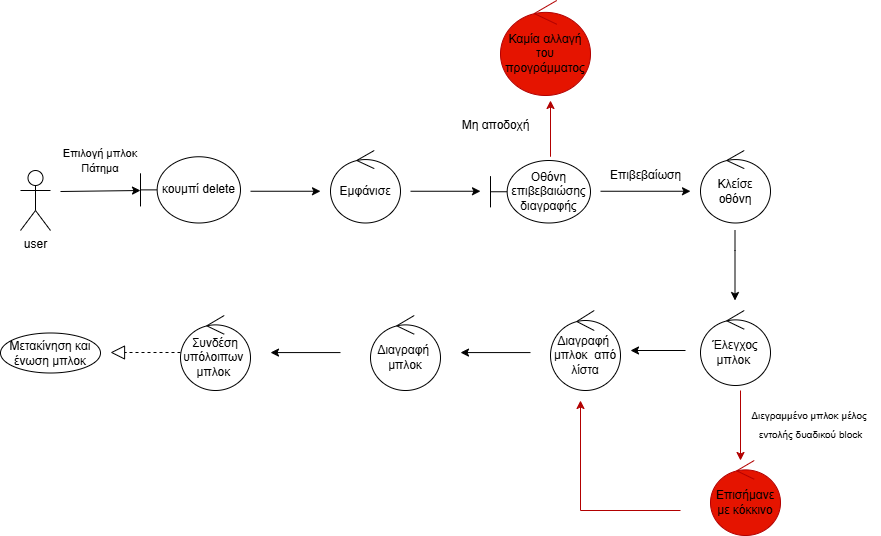
**Σχ5. Διάγραμμα ευρωστίας για Use Case 5**

**Βασική Ροή: Ο χρήστης μετακινεί και ενώνει μπλοκ εντολών.**

Ο χρήστης επιλέγει το μπλοκ που θέλει να μετακινήσει. Κατά την μετακίνηση το σύστημα ελέγχει αν βρίσκεται κοντά σε άλλο μπλοκ και χρωματίζει τη κατάλληλη θέση που αναμένεται να τοποθετηθεί το μπλοκ. Αν υπάρχει μπλοκ σε αυτή την θέση τότε μετακινείται δεξιά. Ο χρήστης τοποθετεί στη επιθυμητή θέση το μπλοκ και το σύστημα το επισυνάπτει στο πιο κοντινό μπλοκ. Έπειτα γίνεται έλεγχος για το αν μπορεί να αποτελεί μέλος της βασικής ροής ή της υπορουτίνας και αναλόγως το σύστημα ενημερώνει τη λίστα με το id του μπλοκ σε κατάλληλη θέση.

**Εναλλακτική Ροή 1: Ο χρήστης μετακινεί το μπλοκ εκτός ορίου καμβά.**

Ο χρήστης προσπαθεί να μετακινήσει το μπλοκ εκτός ορίου καμβά αλλά το σύστημα κρατάει το μπλοκ εντός του επιτρέπων ορίων.



**Σχ6. Διάγραμμα ευρωστίας για Use Case 6**

**Βασική Ροή: Ο χρήστης θέλει να διαγράψει ένα μπλοκ.**

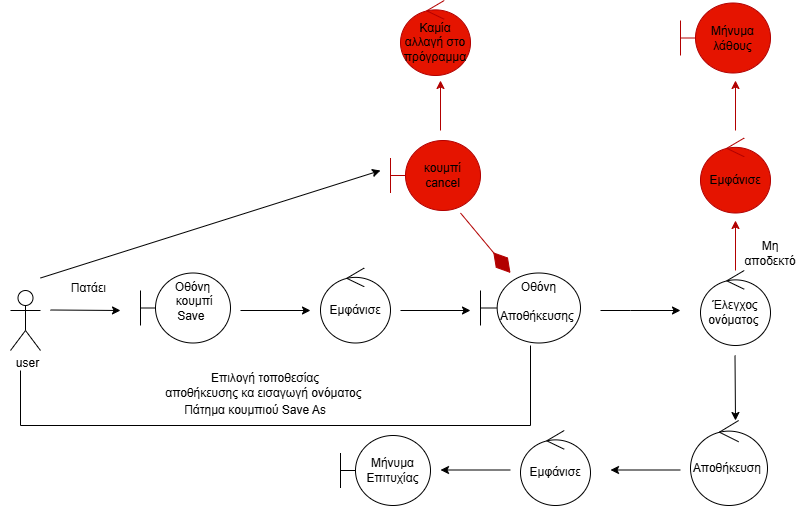
Ο χρήστης επιλέγει το μπλοκ που επιθυμεί να διαγράψει και πατάει το κουμπί Διαγραφής του μπλοκ με αποτέλεσμα να εμφανίζεται παράθυρο επιβεβαίωσης διαγραφής. Ο χρήστης επιβεβαιώνει και κλείνει το παράθυρο. Έπειτα το σύστημα ελέγχει το μπλοκ αν αποτελεί μέλος εντολής δυαδικού μπλοκ και διαγράφει το μπλοκ από την λίστα. Τέλος διαγράφεται το μπλοκ και συνδέονται μεταξύ τους τα εναπομείναντα μπλοκ.

**Εναλλακτική Ροή 1: Ο χρήστης πατάει ακύρωση όταν εμφανίζεται το παράθυρο επιβεβαίωσης.**

Ο χρήστης απορρίπτει το παράθυρο επιβεβαίωσης διαγραφής μπλοκ και δεν υπάρχουν αλλαγές στο πρόγραμμα.

**Εναλλακτική Ροή 2: Ο χρήστης επέλεξε να διαγράψει μπλοκ που αποτελεί μέλος εντολή δυαδικού μπλοκ.**

Το διαγραμμένο μπλοκ αποτελεί μέλος εντολής δυαδικού μπλοκ και επισημαίνεται με κόκκινο στον χρήστη.



**Σχ7. Διάγραμμα ευρωστίας για Use Case 7**

**Βασική Ροή: Ο χρήστης αποθηκεύει το πρόγραμμα του.**

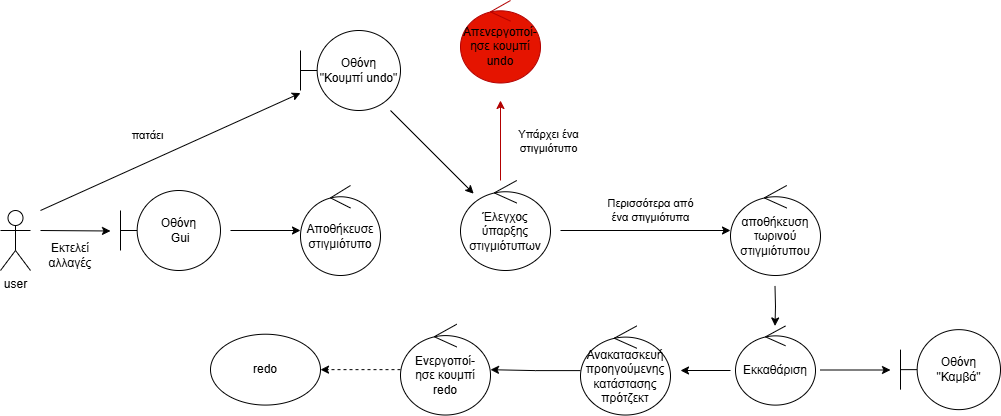
Ο χρήστης αποθηκεύει το πρόγραμμα πατώντας το κουμπί Αποθήκευσης και εμφανίζεται παράθυρο τοποθεσίας για την αποθήκευση του αρχείου του. Ο χρήστης καλείται να εισάγει όνομα στο αρχείο και εφόσον το σύστημα ελέγχει την εγκυρότητα του ονόματος τότε ο χρήστης επιλέγει την τοποθεσία αποθήκευσης που επιθυμεί και με το πάτημα του κουμπιού Save As αποθηκεύεται το αρχείο. Μετά την αποθήκευση ο χρήστης ενημερώνεται με μήνυμα επιτυχίας.

**Εναλλακτική Ροή 1: Ο χρήστης εισάγει μη έγκυρο όνομα στο αρχείο που θέλει να αποθηκεύσει .**

Ο χρήστης βάζει μη έγκυρο όνομα και εμφανίζεται μήνυμα σφάλματος.

**Εναλλακτική Ροή 2: Ο χρήστης πατάει το κουμπί ακύρωσης στο παράθυρο Αποθήκευσης.**

Ο χρήστης δεν επιθυμεί πια την αποθήκευση του αρχείου και δεν γίνεται καμία αλλαγή στο πρόγραμμα.



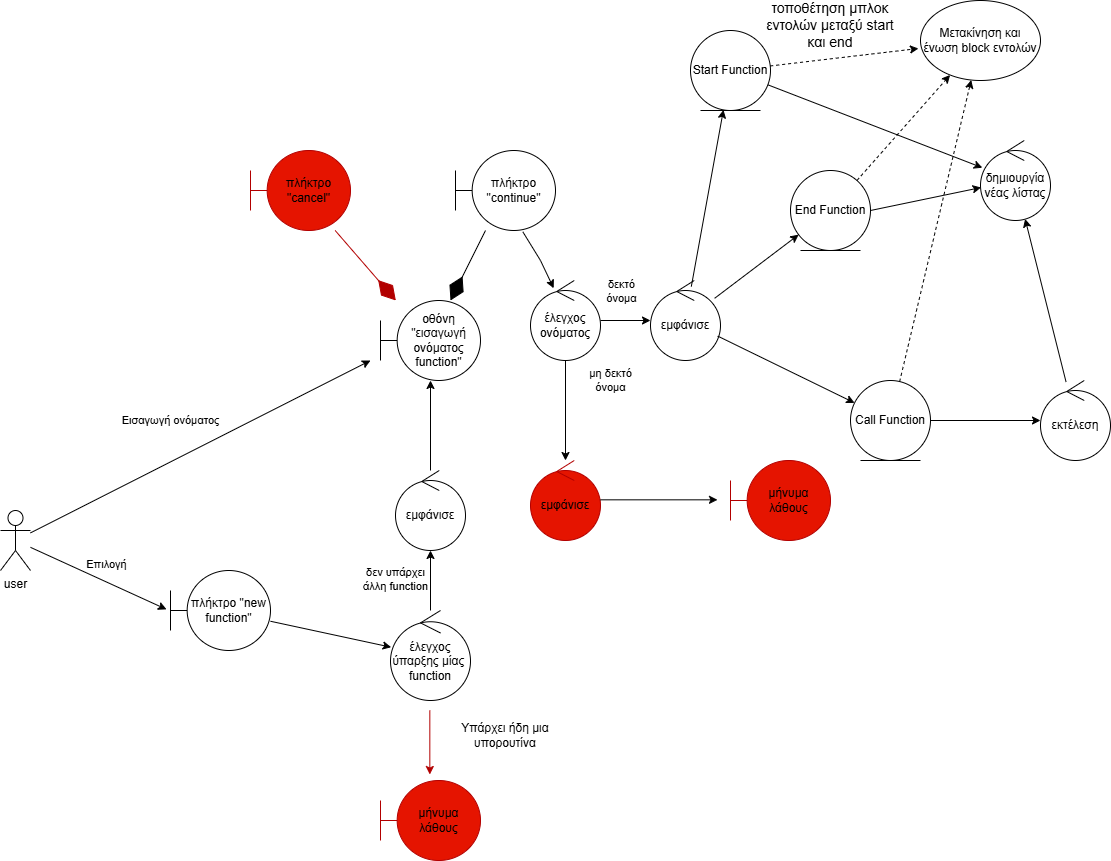
**Σχ8. Διάγραμμα ευρωστίας για Use Case 8**

**Βασική Ροή: Ο χρήστης θέλει να επιστρέψει στην προηγούμενη κατάσταση του πρότζεκτ του.**

Ο χρήστης εκτελεί αλλαγές στο πρότζεκτ του, πατάει το κουμπί undo και γίνεται έλεγχος ύπαρξης στιγμιότυπων. Εφόσον είναι περισσότερα από ένα γίνεται η αποθήκευση του τωρινού στιγμιότυπου από το σύστημα και στην συνέχεια γίνεται εκκαθάριση τη οθόνης. Το σύστημα ανακατασκευάζει την προηγούμενη κατάσταση του πρότζεκτ και ενεργοποιεί το κουμπί redo.

**Εναλλακτική Ροή 1: Υπάρχει ένα μόνο στιγμιότυπο.**

Ο χρήστης πατάει το undo κουμπί ενώ υπάρχει ένα μόνο στιγμιότυπο αποθηκευμένο και το σύστημα απενεργοποιεί το κουμπί undo.



**Σχ9. Διάγραμμα ευρωστίας για Use Case 9**

**Βασική Ροή: Ο χρήστης θέλει να δημιουργήσει μια υπορουτίνα.**

Ο χρήστης δημιουργεί μια νέα υπορουτίνα μέσω του κουμπιού new function και εφόσον δεν υπάρχει άλλη υπορουτίνα εμφανίζεται κενό πεδίο όπου ο χρήστης καλείται να εισάγει όνομα. Ο χρήστης εισάγει όνομα και εμφανίζονται τα μπλοκ start function και end function όπου μεταξύ αυτών πρέπει ο χρήστης να τοποθετήσει μπλοκ εντολών που επιθυμεί και το μπλοκ που να καλεί την υπορουτίνα. Τα μπλοκ της υπορουτίνας αποθηκεύονται σε ξεχωριστή λίστα από αυτή του βασικού προγράμματος και στην συνέχεια βάζει το μπλοκ κλήσης στην λίστα της βασικής ροής.

**Εναλλακτική Ροή 1: Ο χρήστης προσπαθεί να δημιουργήσει υπορουτίνα ενώ ήδη έχει δημιουργήσει μία.**

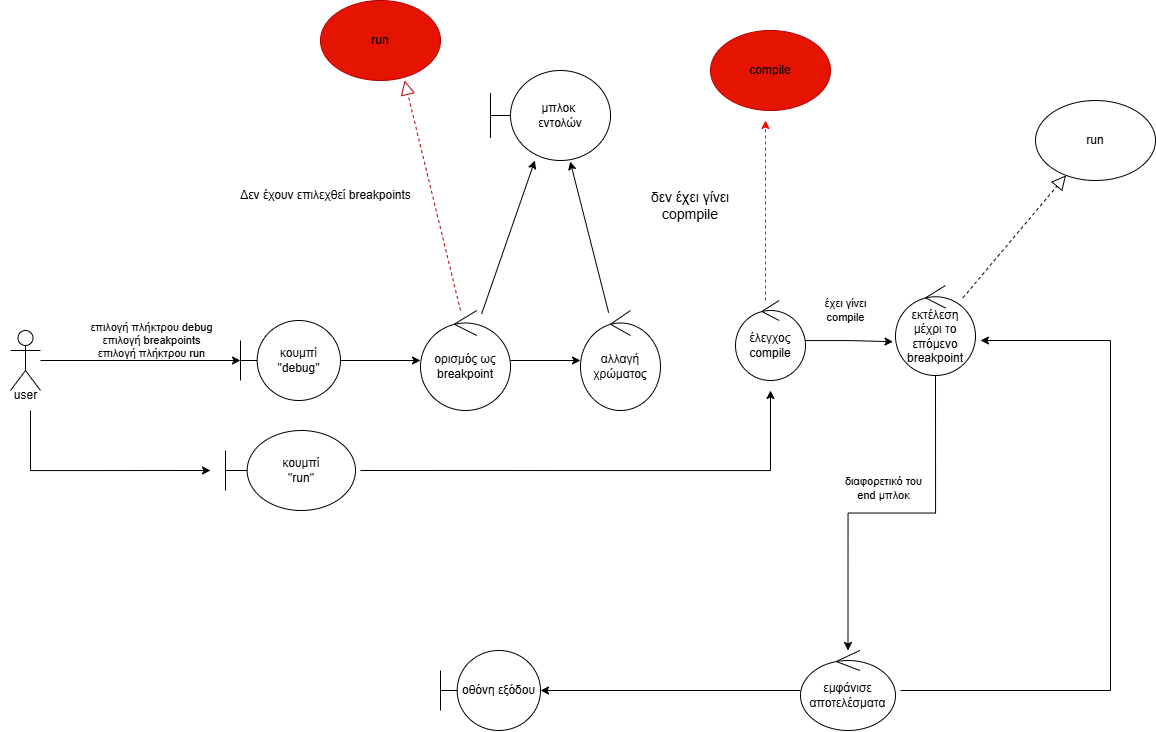
Όταν γίνεται ο έλεγχος ύπαρξης μίας υπορουτίνας, το σύστημα αντιλαμβάνεται ότι υπάρχει ήδη μια υπορουτίνα και εμφανίζει μήνυμα σφάλματος στον χρήστη.

**Εναλλακτική Ροή 2: Ο χρήστης εισάγει μη αποδεκτό όνομα στην υπορουτίνα.**

Ο χρήστης εισάγει μη αποδεκτό όνομα στην υπορουτίνα και εμφανίζεται μήνυμα σφάλματος.

**Εναλλακτική Ροή 3: Ο χρήστης ακυρώνει την διαδικασία δημιουργίας νέας υπορουτίνας.**

Ο χρήστης ακυρώνει την διαδικασία δημιουργίας νέας υπορουτίνας και το πρόγραμμα παραμένει ίδιο.

****

**Σχ10. Διάγραμμα ευρωστίας για Use Case 10**

**Βασική Ροή: Ο χρήστης κάνει debug το πρόγραμμα του.**

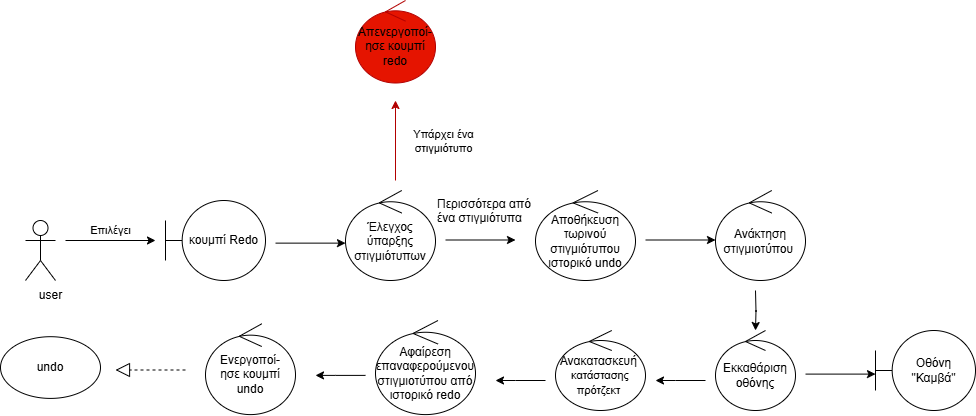
Ο χρήστης επιλέγει το κουμπί του debug και τα μπλοκ εντολών που θα χρησιμοποιήσει ως breakpoints, τα οποία και χρωματίζονται. Το σύστημα τότε ενεργοποιεί το debug mode και ο χρήστης εκτελεί το πρόγραμμα. Το σύστημα ελέγχει αν έχει γίνει compile πριν την τελευταία αλλαγή και το πρόγραμμα εκτελείται μέχρι το πρώτο breakpoint ενημερώνοντας την οθόνη εξόδου με τις νέες τιμές. Έπειτα το σύστημα ελέγχει για την ύπαρξη επόμενου breakpoint και εφόσον υπάρχει, ο χρήστης πατάει ξανά run με αποτέλεσμα να επαναλαμβάνονται τα ίδια βήματα όπως και στο πρώτο breakpoint μέχρι να τερματίσει το πρόγραμμα και να απενεργοποιηθεί το debug mode.

**Εναλλακτική Ροή 1: Ο χρήστης δεν έχει κάνει compile πριν την τελευταία αλλαγή του προγράμματος.**

Το σύστημα διαπιστώνει ότι δεν έχει γίνει compile και ενημερώνει τον χρήστη με μήνυμα σφάλματος.

**Εναλλακτική Ροή 2: Ο χρήστης δεν επιλέγει breakpoints για το πρόγραμμα του.**

Το σύστημα εκτελεί όλο το πρόγραμμα.

****

**Σχ11. Διάγραμμα ευρωστίας για Use Case 11**

**Βασική Ροή: Ο χρήστης θέλει να ανακτήσει την κατάσταση του πρότζεκτ που αναιρέθηκε τελευταία.**

Ο χρήστης πατάει το κουμπί redo και το σύστημα κάνει έλεγχο για την ύπαρξη στιγμιότυπων. Εφόσον υπάρχουν περισσότερα από ένα, γίνεται η αποθήκευση του τωρινού στιγμιότυπου στο ιστορικό undo και γίνεται η ανάκτηση του στιγμιότυπου που επιθυμεί ο χρήστης. Αφού γίνει η εκκαθάριση οθόνης και η ανακατασκευή του πρότζεκτ, αφαιρείται το στιγμιότυπο από ιστορικό redo και ενεργοποιείται το κουμπί redo.

**Εναλλακτική Ροή 1: Υπάρχει ένα μόνο στιγμιότυπο.**

Ο χρήστης πατάει το redo κουμπί ενώ υπάρχει ένα μόνο στιγμιότυπο αποθηκευμένο και το σύστημα απενεργοποιεί το κουμπί redo.