

Ανάλυση του Συστήματος της Εφαρμογής

Εισαγωγή

Ο στόχος αυτής της εφαρμογής είναι να σχεδιαστεί ένα σύστημα που να προσομοιώνει «έναν προσωπικό βοηθό». Το front-end του συστήματος είναι μια δυναμική σελίδα η οποία πολύ περιεκτικά θα παρουσιάζει όλες τις λειτουργίες του συστήματος και θα αποτελεί την γέφυρα επικοινωνίας μεταξύ του χρήστη και του συστήματος. Το σύστημα θα είναι ανοικτό συνέχεια ώστε να βοηθήσει αμέσως και θα είναι συνδεδεμένο σε μία βάση δεδομένων ώστε να μπορεί να αποθηκεύει ότι χρειαστεί. Δικαίωμα χρήσης του συστήματος θα την έχει μόνο ένας ο χρήστης(π.χ. το σύστημα θα αναγνωρίζει την φωνή του, ή θα του ζητάει μοναδικό κωδικό, κ.α.). Το σύστημα θα πρέπει να επιτρέπει στους χρήστες να επιλέξουν τις λειτουργίες που επιθυμούν να εκτελέσουν και να βλέπουν τα αποτελέσματα, σε κάθε τους βήμα ο βοηθός θα του παρέχει φωνητικές εντολές που θα τους κατευθύνει και θα τους βοηθάει.

1.1 Τεχνολογία Ανάπτυξης Λογισμικού

Ως σύστημα ορίζεται μια ομάδα ή ένα σύνολο αντικειμένων μεταξύ των οποίων υπάρχει αντίδραση, αλληλεπίδραση και αλληλεξάρτηση έτσι ώστε να δημιουργείται μία ενιαία οντότητα η οποία διαμορφώνεται βάση των συνθηκών του περιβάλλοντος μέσα στο οποίο υπάρχει και η δημιουργία του συνόλου αυτού έχει να επιτελέσει κάποιο σκοπό. Παντού γύρω μας υπάρχουν συστήματα διαφόρων τύπων τα οποία αλληλοεπιδρούν και αλληλεξαρτώνται. Κάθε σύστημα μπορεί να στηρίζεται σε ένα σύνολο μικρότερων συστημάτων τα οποία ονομάζονται υποσυστήματα. Η χρήση των ηλεκτρονικών υπολογιστών και η εξάπλωση του διαδικτύου είναι ραγδαία με αποτέλεσμα να έχουν δημιουργηθεί νέες καταστάσεις και νέα δεδομένα. Αυτό έχει ως αποτέλεσμα να επιβάλλεται η δημιουργία κατάλληλων μοντέλων τα οποία έχουν διατυπωθεί μέσω εφαρμογών για την ταχύτατη επίλυση προβλημάτων τα οποία για τον άνθρωπο θα απαιτούσαν πολύ περισσότερο χρόνο. Έτσι λοιπόν ο ηλεκτρονικός υπολογιστής αποτελεί ένα σύστημα το οποίο και αυτό με την σειρά του αποτελείται από άλλα συστήματα όπως το υλικό και το λογισμικό. Το λογισμικό αποτελεί το σύνολο των προγραμμάτων, δηλαδή μια αλληλουχία οδηγιών προς το σύστημα για την εκτέλεση κάποιας συγκεκριμένης διεργασίας, καθώς και των εφαρμογών τις οποίες μπορεί να αναπτύξει ο χρήστης. (Βεσκούδης Β., 2000)

Φτάνοντας στο σημείο της δημιουργίας εφαρμογών από των χρήστη αναπτύσσεται μια νέα επιστήμη αυτή της τεχνολογίας λογισμικού η οποία ασχολείται με την αναζήτηση και θέσπιση μεθόδων για την περιγραφή κατασκευή και συντήρηση του λογισμικού. Έτσι λοιπόν έχουν αναπτυχθεί κάποιες μέθοδοι ή αλλιώς μοντέλα του κύκλου ζωής λογισμικού. (Χατζόγλου, 2005: 24)

Οι βασικοί πυλώνες που διαθέτουν όλα τα μοντέλα είναι οι εξής:

- Η σύλληψη της ιδέας
- Η καταγραφή των απαιτήσεων
- Η ανάλυση των απαιτήσεων του συστήματος
- Η σχεδίαση της εφαρμογής
- Η ανάπτυξη του κώδικα
- Ο έλεγχος
- Η χρήση και συντήρηση
- Η απόσυρση

Επομένως, το μοντέλο κύκλου ζωής λογισμικού είναι η περιγραφή των μεθόδων και των διαδικασιών που ακολουθούνται σε κάθε φάση της ανάπτυξης του λογισμικού από την σύλληψη ως την απόσυρση του λογισμικού. (Χατζόγλου, 2005: 27-28).

1.2 Καταγραφή απαιτήσεων

Η ακριβής και σωστή καταγραφή των απαιτήσεων είναι πολύ σημαντική για την περαιτέρω ανάπτυξη της εφαρμογής. Παρακάτω θα δούμε αναλυτικά τις απαιτήσεις:

- Να ελέγξει τις ηλεκτρικές συσκευές του σπιτιού.
- Να ελέγξει τις τα φώτα του σπιτιού.
- Να δει και να προγραμματίσει το καθημερινό του πρόγραμμα.
- Να ελέγξει την κατάσταση, το νερό και το φαγητό του κατοικίδιού του.
- Να κάνει καθημερινές εργασίες στον υπολογιστή(να δει την παπουτσοθήκη του, να ακούσει η μουσική, να δει τον καιρό, να κάνει αναζήτηση στο ίντερνετ, να παραγγείλει να φάει, να διαβάσει εφημερίδα ή τα e-mail, να δει την κίνηση στους δρόμους, κ.α.)
- Να λάβει κάποια βοήθεια σε περίπτωση που θα τη χρειαστεί.

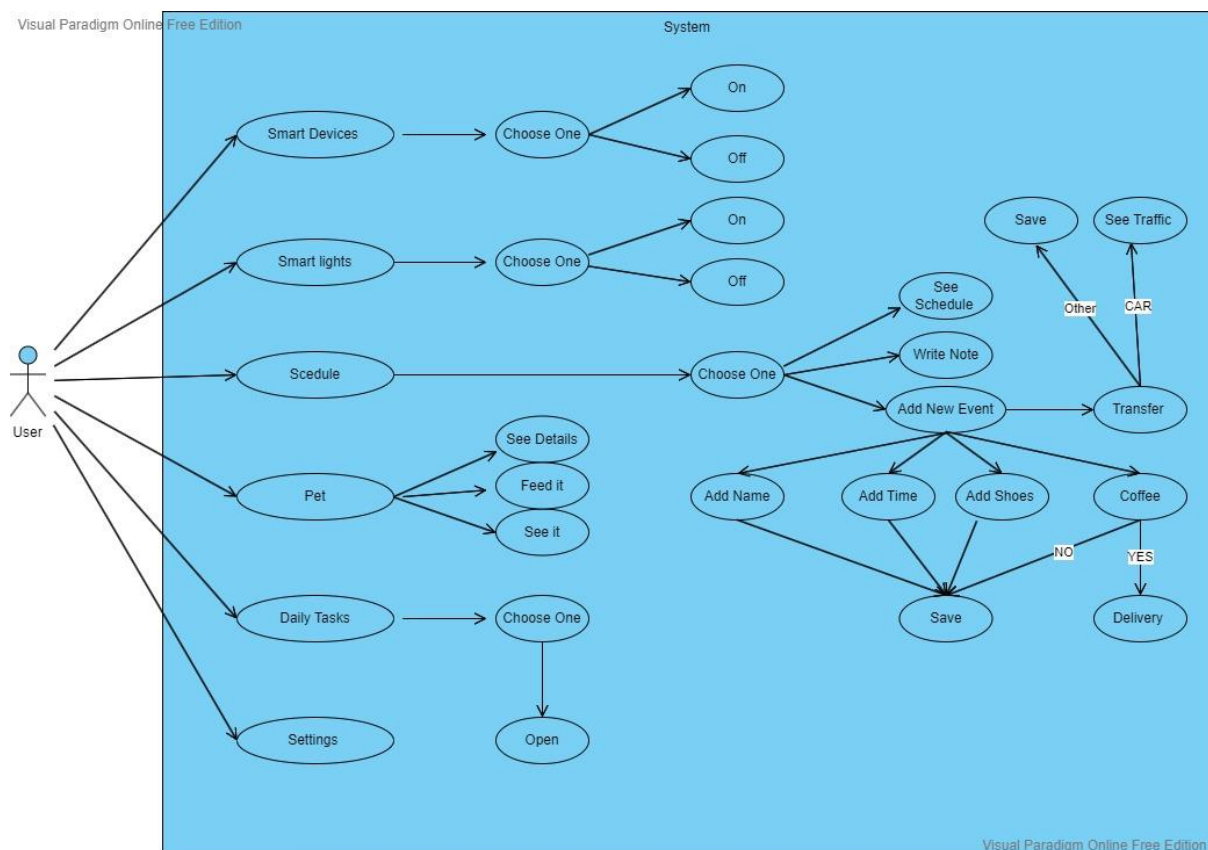
Για να προσθέσει μια καινούργια εκδήλωση στο πρόγραμμά του ο χρήστης θα χρειαστεί να πληκτρολογήσει:

- Το όνομά της.
- Την ώρα που θα πραγματοποιηθεί.
- Τι παπούτσια θα φορέσει.
- Τη διεύθυνση που θα πάει.
- Αν θα χρειαστεί καφέ.
- Με ποιο μέσο θα πραγματοποιήσει τη μεταφορά του σε αυτήν.

1.3 Ανάλυση απαιτήσεων

Διάγραμμα Περίπτωσης Χρήσης (Use Case Diagram) της εφαρμογής μας

Ένα διάγραμμα περίπτωσης χρήσης (use case diagram) είναι ένα δυναμικό διάγραμμα ή διάγραμμα συμπεριφοράς στην UML. Τα διαγράμματα περιπτώσεων χρήσης μοντελοποιούν τη λειτουργικότητα ενός συστήματος χρησιμοποιώντας φορείς και περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης είναι ένα σύνολο ενεργειών, υπηρεσιών και λειτουργιών που πρέπει να εκτελεί το σύστημα. Σε αυτό το πλαίσιο, ένα "σύστημα" είναι κάτι που αναπτύσσεται ή λειτουργεί, όπως ένας δικτυακός τόπος. Οι "φορείς" είναι άτομα ή οντότητες που λειτουργούν υπό καθορισμένους ρόλους εντός του συστήματος. Τα διαγράμματα περιπτώσεων χρήσης είναι πολύτιμα για την οπτικοποίηση των λειτουργικών απαιτήσεων ενός συστήματος που θα μεταφραστούν σε επιλογές σχεδιασμού και προτεραιότητες ανάπτυξης. Βοηθούν επίσης στον εντοπισμό τυχόν εσωτερικών ή εξωτερικών παραγόντων που μπορεί να επηρεάσουν το σύστημα και θα πρέπει να ληφθούν υπόψη. Παρέχουν μια καλή ανάλυση υψηλού επιπέδου από το εξωτερικό του συστήματος. Τα διαγράμματα περιπτώσεων χρήσης καθορίζουν τον τρόπο με τον οποίο το σύστημα αλληλοεπιδρά με τους φορείς χωρίς να ανησυχούν για τις λεπτομέρειες του τρόπου με τον οποίο υλοποιείται αυτή η λειτουργικότητα.



Σχεδίαση της εφαρμογής

Για την υλοποίηση της εφαρμογής θα χρησιμοποιήσουμε το Visual Studio και τη γλώσσα προγραμματισμού C#. Ο κώδικας της εφαρμογής καθώς και η υλοποίηση της θα παραδοθεί σε ξεχωριστό αρχείο. Με το άνοιγμα της εφαρμογής θα μπορούμε να περάσουμε τα δεδομένα της μέρας, η εφαρμογή θα παραμένει ανοικτή και ο προσωπικό βοηθός θα μας ενημερώνει για τις εξελίξεις.

Σε περίπτωση όμως που θέλουν τα δεδομένα μας να αποθηκεύονται, για την καλύτερη υλοποίηση του προγράμματος θα πρέπει να είναι συνδεδεμένο με μια βάση δεδομένων. Η βάση αυτή θα αποτελείτε από 2 πίνακες. Ο ένας θα καταγράφει τις εκδηλώσεις και τις διάφορες λεπτομέρειες που θα πρέπει με περιέχονται στον πίνακα αυτόν όπως το όνομα της εκδήλωσης, η ώρα, τα παπούτσια που θα χρειαστούμε, τη διεύθυνση και το μεταφορικό μέσο που θα χρειαστούμε για να πάμε. Στον δεύτερο πίνακα θα αποθηκεύαμε τις ημέρες και τι εκδηλώσεις έχουμε να πραγματοποιήσουμε κάθε μέρα. Οι πίνακες αυτοί θα μπορούσαν να συνδεθούν(Primary Key) είτε μέσω της μεταβλητής ώρας είτε μέσω του ονόματος της κάθε μεταβλητής.

Μοντέλο διάδρασης χρήστη-συστήματος κατά Norman

Το μοντέλο του Norman είναι ένα από τα πιο σημαντικά μοντέλα αλληλεπίδρασης στην αλληλεπίδραση ανθρώπου υπολογιστή. Ο Norman κατασκεύασε το μοντέλο αυτό για να δείξει γιατί δημιουργούνται προβλήματα στους χρήστες κατά την εκτελεστεί μιας διεπαφής και το χώρισε στα εξής στάδια:

1. Τίθεται ο στόχος.
2. Καθορισμός της πρόθεσης.
3. Καθορισμός της σειράς ενεργειών.
4. Εκτέλεση της ενέργειας.
5. Αντίληψη της κατάστασης του συστήματος
6. Ερμηνεία της κατάστασης του συστήματος.
7. Αξιολόγηση της κατάστασης του συστήματος σε σχέση με τους στόχους και τις προθέσεις.

Για την εφαρμογή του μοντέλου θα πρέπει να δημιουργήσουμε ένα υποθετικό σενάριο.

Πρώτο στάδιο: Έστω ότι ο χρήστης θέλει να ανοίξει την πόρτα του γκαράζ.

Δεύτερο στάδιο: Ο χρήστης μας(π.χ. η Αριάδνη) καθορίζει την πρόθεσή του να χρησιμοποιήσει την εφαρμογή μας και να ανοίξει την πόρτα του γκαράζ.

Τρίτο στάδιο: Η Αριάδνη είναι μπροστά στον υπολογιστή.

1. Θα πρέπει να ανοίξει την εφαρμογή.
2. Να ψάξει είχε την εικόνα του γκαράζ.
3. Να κάνει κλικ στην αντίστοιχη εικόνα.

Τέταρτο στάδιο: Με το άνοιγμα της εφαρμογής ο έξυπνος οικιακός βοηθός θα την κατατοπίσει με φωνητικές εντολές και θα τη βοηθήσει να βρει την εικόνα, κάνοντας κλικ θα μπορέσει να ανοίξει ή να κλείσει την πόρτα.

Πέμπτο στάδιο: Η Αριάδνη με το που ανοίξει την εφαρμογή θα ακούσει τις φωνητικές εντολές και θα δει ότι η πόρτα του γκαράζ είναι κλειστή αφού η ένδειξη κάτω από την εικόνα είναι κόκκινη.

Έκτο στάδιο: Μετά το κλικ πάνω στην εικόνα η ένδειξη θα γίνει πράσινη και θα αντιληφθεί ότι η πόρτα άνοιξε.

Έβδομο στάδιο: Η Αριάδνη θα αξιολογήσει την κατάσταση σε σχέση με τον στόχο. Η εφαρμογή θα δείχνει ότι έχει πάρει την εντολή να ανοίξει η πόρτα και θα είναι ευχαριστημένη γιατί εκτέλεσε με επιτυχία την ενέργεια που ήθελε μέσω της εφαρμογής μας.

Κανόνες καλού σχεδιασμού

Κατά τη σχεδίαση του συστήματος διεπαφής δόθηκε ιδιαίτερη σημασία στο να τηρούνται οι κανόνες καλού σχεδιασμού.

1. **Συνέπεια:** Το πρόγραμμά μας έχει καλή δομή είναι χωρισμένο σε 5 μέρη τα οποία είναι ευκρινώς διαχωρίσιμα, πράγμα που κάνουν τη χρήση του αρκετά εύκολη. Τα εικονίδια που χρησιμοποιούμε βοηθούν στην κατανόηση της κάθε ενέργειας και χρησιμοποιούμε τα χρώματα μας κατάλληλα είτε για να δώσουμε κάποιο στοιχείο στο χρήστη είτε για να δημιουργήσουμε το μενού μας.
2. **Shortcuts:** Οποιοσδήποτε χρήστης μπορεί να χρησιμοποιεί το παρακάτω πρόγραμμα ανεξαρτήτως φύλου, ηλικίας και μορφωτικού επιπέδου.
3. **Περιεκτικά μηνύματα:** Τα μηνύματα που χρησιμοποιούμε για τη διευκόλυνση του χρήστη είναι περιεκτικά και δίνουν την πληροφορία που χρειάζεται με συνοπτικό τρόπο.
4. **Σχεδιασμός διαλόγων με πληρότητα:** Οι διάλογοί μας είναι συνοπτικοί και πλήρης.
5. **Αποφυγή και εύκολη διαχείριση λαθών:** Ο σχεδιασμός της εφαρμογής είναι επίτηδες εύκολος ώστε να αποτρέπει τα λάθη, σε κάθε περίπτωση υπάρχει φωνητική εντολή και αντίστοιχο μήνυμα ώστε να βοηθάει το χρήστη να επιτύχει το στόχο του.
6. **Δυνατότητα επαναφοράς της κατάστασης πριν από την ενέργεια:** Η εφαρμογή διαθέτει αντίστοιχο κουμπί(κουμπί Settings) που βοηθάει στην αναίρεση της κάθε κίνησης του χρήστη.
7. **Πρωτοβουλία του ελέγχου στο χρήστη:** Κάθε χρήστης στην περίπτωσή μας έχει τον έλεγχο της εφαρμογής και ο βοηθός βρίσκεται εκεί για να τον συμβουλέψει.
8. **Μείωση του φόρτου της μνήμης της μικρής διάρκειας:** Ο χρήστης με το που ανοίξει την εφαρμογή δεν χρειάζεται να έχει ξαναδεί κάποια αντίστοιχη εφαρμογή ή να έχει κάποια προ υπάρχουσα γνώση και ο βοηθός βρίσκεται συνέχεια σε εγρήγορση για να του την προσφέρει.