



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πράσινες ΤΠΕ - Ολοκληρωμένο Σύστημα
Αυτόματου Ελέγχου Κτιρίων**

Η χρήση των υπηρεσιών Skype στα ολοκληρωμένα συστήματα
Αυτομάτου ελέγχου κτιρίων

Προπτυχιακός Φοιτητής: Γκατζιούρας Εμμανουήλ
ΑΜ:4116

Διπλωματική Εργασία

Επιβλέπων: Καθηγητής Χρήστος Ζαρολιάγκης

Συνεπιβλέπων: Δρ. Ιωάννης Χατζηγιαννάκης

Πάτρα Μάρτιος 2013

Πρόλογος

Περίληψη

Το παρόν κείμενο αποτελεί τη διπλωματική εργασία του συγγραφέα για το ακαδημαϊκό έτος 2011-1012. Παρουσιάζεται το έργο SkypeDust, μια εφαρμογή προσωπικού υπολογιστή η οποία αναπτύχθηκε για την εξυπηρέτηση χρηστών ολοκληρωμένων συστημάτων ελέγχου κτηρίων. Το λογισμικό SkypeDust αλληλεπιδρά με το ολοκληρωμένο σύστημα ελέγχου Überdust. Σε αυτήν την διπλωματική παρουσιάζονται οι στόχοι ,οι προκλήσεις του έργου, οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν με κύρια αναφορά στην προγραμματιστική διεπαφή του Skype, και τα εργαλεία που παρέχει η γλώσσα προγραμματισμού Java. Επίσης παρουσιάζονται οι υπηρεσίες που αναπτύχθηκαν και οι μελλοντικές κατευθύνσεις του έργου.

Ευχαριστίες

Αρχικά, ευχαριστώ τους καθηγητές μου κ. Χρήστο Ζαρολιάγκη και κ. Ιωάννη Χατζηγιαννάκη, τόσο για την διεκπεραίωση της παρούσας εργασίας όσο και για την πολύτιμη καθοδήγηση που μου παρείχαν τα τελευταία χρόνια των σπουδών μου. Τους είμαι ιδιαίτερα ευγνώμων για τις συμβουλές που μου έδωσαν και για τις γνώσεις που κέρδισα από την πολυετή εμπειρία τους στο χώρο της Πληροφορικής.

Επίσης, θα ήθελα να ευχαριστήσω τους Ορέστη Ακριβόπουλο και Δημήτριο Αμαξιλάτη για την βοήθεια και συνεργασία κατά την εκπόνηση αυτής της διπλωματικής εργασίας.

Ακόμη ευχαριστώ τους φίλους και συνεργάτες Φώτη Στεφανίδη και Κωσταντίνο Ακάσογλου .

Τέλος Ευχαριστώ την οικογένειά μου για την υλική και ψυχολογική υποστήριξη που μου παρείχαν.

Γκατζιούρας Εμμανουήλ

Περιεχόμενα

1	Εισαγωγή	5
1.1	Εισαγωγή στα Ολοκληρωμένα Συστήματα Ελέγχου Κτηρίων	5
1.2	Υπάρχοντα Συστήματα και το σύστημα Überdust	8
1.2.1	Το Σύστημα Überdust	9
1.3	Στόχοι Διπλωματικής Εργασίας	11
1.4	Συνεισφορά Διπλωματικής Εργασίας	11
1.5	Οργάνωση Διπλωματικής Εργασίας	13
2	Η χρήση υπηρεσιών Skype	15
2.1	Το λογισμικό Skype	15
2.2	Εισαγωγή στη προγραμματιστική χρήση υπηρεσιών Skype	16
3	Η εφαρμογή SkypeDust	19
3.1	Γενική αρχιτεκτονική συστήματος SkypeDust	19
3.2	Περιγραφή της εφαρμογής	20
3.2.1	Διαχείριση Λογαριασμού	20
3.2.2	Ονοματολογία κόμβων και δυνατοτήτων	21
3.2.3	Διαχείριση plugins	24
3.3	Εκτέλεση της εφαρμογής	26
3.3.1	Προεπιλεγμένη εκτέλεση εφαρμογής	26
3.3.2	Εκτέλεση εφαρμογής με χρήση πρόσθετου λογισμικού . . .	27
4	Η υλοποίηση του SkypeDust	29
4.1	Περιγραφή αρχιτεκτονικής	29
4.2	Ο Διαχειριστής SkypeDust	30
4.3	Η Διεπαφή Χρήστη	32
4.4	Ο Διαχειριστής Ρυθμίσεων	34
4.5	Ο Χρήστης Skype	36
4.6	Ο Πελάτης Überdust	37
4.7	Ο Διαχειριστής Πρόσθετου Λογισμικού	39
4.8	Ο Πάροχος Δεδομένων	40

4.8.1	Τρόποι Αποθήκευσης Δεδομένων	40
4.8.2	Η βάση δεδομένων του SkypeDust	41
4.9	Ροή Λειτουργίας	44
5	Χρήση πρόσθετου λογισμικού	47
5.1	Προϋποθέσεις	47
5.1.1	Γενικές Προϋποθέσεις	47
5.2	Οι κατηγορίες	48
5.2.1	Διαχείριση Μηνυμάτων	49
5.2.2	Διαχείριση Restfull Πελάτη	50
5.2.3	Διαχείριση Ασύγχρονων Ενημερώσεων	50
5.2.4	Παραδείγματα Πρόσθετου Λογισμικού	51
6	Συμπεράσματα και προοπτικές	53
6.1	Συμπεράσματα	53
6.2	Μελλοντικοί Στόχοι	55

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στα Ολοκληρωμένα Συστήματα Ελέγχου Κτηρίων

Τα τελευταία χρόνια τα συστήματα ελέγχου κτηρίων [5] έχουν κερδίσει το ενδιαφέρον του χώρου της πληροφορικής. Στον ακαδημαϊκό χώρο λόγω θεωρητικού υπόβαθρου που απαιτείται, αλλά και στη βιομηχανία λόγω των πρακτικών εφαρμογών που μπορούν να προκύψουν.

Η εγκατάσταση ενός συστήματος ελέγχου σε ένα κτήριο εξασφαλίζει τη συλλογή δεδομένων που αφορούν μετρήσεις στοιχείων του περιβάλλοντος, αλλά και πληροφορίες για τη κατάσταση των συσκευών που είναι τοποθετημένες στο κτήριο. Επίσης δίνεται η δυνατότητα αντίδρασης και εκτέλεσης ενεργειών ανάλογα με τα δεδομένα όπως και η απευθείας αλληλεπίδραση με τις συσκευές του κτηρίου για παράδειγμα ένα κλιματιστικό.

Το εύρος των εφαρμογών είναι τεράστιο. Μπορούμε να έχουμε εφαρμογές που ενημερώνουν για καταστάσεις του περιβάλλοντος όπως η θερμοκρασία η υγρασία και άλλα χαρακτηριστικά, εφαρμογές που μπορούν να επικοινωνήσουν με συσκευές του κτηρίου μέχρι και εφαρμογές συνεχούς ελέγχου και δράσης σε συγκεκριμένες καταστάσεις.

Οι περιπτώσεις χρήσης είναι πολλές, ένα παράδειγμα είναι η περίπτωση ελέγχου και ρύθμισης του φωτισμού και του κλιματισμού για την περίπτωση ατόμων ευπαθών ομάδων τα οποία δεν μπορούν από μόνα τους να προβούν σε κάποιες ενέργειες. Μπορούν να χρησιμοποιηθούν για τον έλεγχο σε περίπτωση ατυχών περιπτώσεων όπως η ενημέρωση αύξησης θερμοκρασίας λόγω φωτιάς ή ανθρώπινα σφάλματα όπως το να αφεθούν ηλεκτρικές συσκευές ανοιχτές. Μια άλλη περίπτωση είναι η παρακολούθηση και ο έλεγχος ενός κτηρίου για την περίπτωση προστασίας από ανεπιθύμητους.

Η αρχιτεκτονική αυτών των συστημάτων περιλαμβάνει ένα ασύρματο δίκτυο

αισθητήρων [37]. Το ασύρματο δίκτυο αισθητήρων αποτελείται από ένα μεγάλο αριθμό τυχαίων και αραιά τοποθετημένων, αυτόνομων αισθητήρων περιορισμένων πόρων και υπολογιστικής ισχύος. Οι παραπάνω συσκευές μπορούν να αναγνωρίσουν χαρακτηριστικά του περιβάλλοντος και να συλλέξουν χρήσιμη πληροφορία όσον αφορά το κτήριο.

Οι κόμβοι έχουν μεταξύ τους ασύρματες μη-κατευθυνόμενες συνδέσεις. Πέρα από την συλλογή των δεδομένων από τους αισθητήρες, σκοπός τους είναι η διάδοση των δεδομένων σε ένα κύριο κόμβο-πύλη (Gateway Node) συνήθως με συνεργατικό τρόπο. Συχνά, ο κόμβος-πύλη είναι αυτός που επικοινωνεί με ένα υπολογιστικό σύστημα το οποίο επεξεργάζεται, διαχειρίζεται ή αποθηκεύει τα δεδομένα που συλλέγει το δίκτυο.

Οι συσκευές αισθητήρων (κόμβοι - Nodes) φέρουν έναν ή περισσότερους αισθητήρες. Μια τυπική αρχιτεκτονική ενός κόμβου αποτελείται από ένα σύστημα πομπού/δέκτη, ένα μικροελεγκτή [22], ένα σύστημα διεπαφής του μικροελεγκτή με τους αισθητήρες και συνήθως μια μπαταρία για πηγή ενέργειας. Το μέγεθος ενός κόμβου κυμαίνεται από ένα σύγχρονο υπολογιστή τσέπης έως το μέγεθος μιας καρφίτσας.



Σχήμα 1.1: Συσκευές με αισθητήρες

Το μέγεθος και το κόστος καθορίζονται από τη λειτουργικότητα που επιθυμεί να έχει ο σχεδιαστής, η οποία εξαρτάται από την υπολογιστική ισχύ, την ταχύτητα τηλεπικοινωνίας και κατανάλωση την ενέργειας της συσκευής.

Τέλος απαραίτητο συστατικό στη λειτουργία ενός κόμβου είναι το λογισμικό που εκτελείται στο μικροελεγκτή του. Οι δυνατότητες του κόμβου εξαρτώνται σε μεγάλο βαθμό από το λογισμικό που εκτελείται στο μικροελεγκτή του. Με το κατάλληλο λογισμικό μπορεί να παρέχεται λειτουργικότητα όπως δρομολόγηση, κρυπτογράφηση, μετρικές δικτύωσης. Συχνά, ο κόμβος-πύλη είναι αυτός που

επικοινωνεί με ένα υπολογιστικό σύστημα το οποίο επεξεργάζεται, διαχειρίζεται και αποθηκεύει τα δεδομένα που συλλέγει το ασύρματο δίκτυο αισθητήρων.

Η επιτυχία αυτών των συστημάτων εξαρτάται σε μεγάλο βαθμό από την παρουσίαση των δεδομένων και την επιλογή λειτουργιών από τον τελικό χρήστη. Πρέπει η επικοινωνία με το δίκτυο συσκευών που είναι εγκατεστημένες στο κτήριο, να είναι άμεση, εύκολη και φιλική προς τον χρήστη. Ο χρήστης θα πρέπει να χρησιμοποιεί λογισμικό με το οποίο να είναι εξοικειωμένος, λογισμικό το οποίο να είναι εύκολο στη χρήση για να είναι εύκολη, η άμεση εκτέλεση ενεργειών. Επιπλέον θα πρέπει να είναι δυνατό ο χρήστης να μπορεί να έχει άμεση και αξιόπιστη πρόσβαση στο σύστημα. Τα δεδομένα πρέπει να είναι έγκυρα και οι εντολές που δίνονται θα πρέπει να εκτελούνται απευθείας.

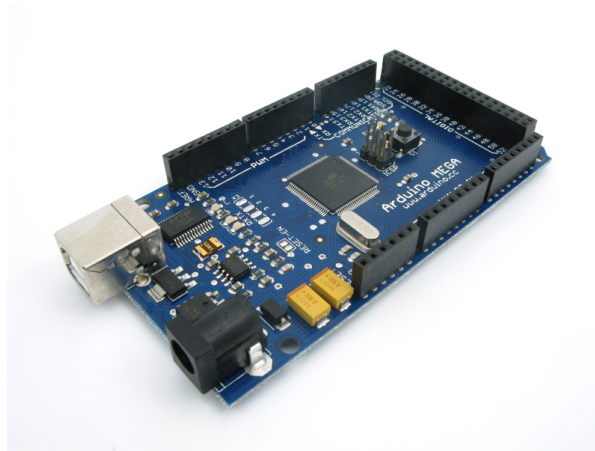
Για να μπορέσουν να επιτευχθούν τα παραπάνω ένα σύστημα ελέγχου πρέπει να είναι ευέλικτο όσο αφορά την παροχή των υπηρεσιών του αλλά και την αλληλεπίδραση με τις εφαρμογές που χρησιμοποιεί ο τελικός χρήστης. Αυτό μπορεί να εξασφαλιστεί με την δημιουργία προγραμματιστικών διεπαφών βασισμένων σε δημοφιλή πρωτόκολλα επικοινωνίας όπως το Πρωτόκολλο Ελέγχου Μετάδοσης (TCP [33]) ή το Πρωτόκολλο Μεταφοράς Υπερκειμένου (Http [14]). Δηλαδή πρωτόκολλα τα οποία υποστηρίζονται από τα περισσότερα εργαλεία που μπορεί να χρησιμοποιεί ένας προγραμματιστής εφαρμογών.

Λόγω αυτών των διεπαφών δίνεται η δυνατότητα στον προγραμματιστή εφαρμογών να χρησιμοποιήσει τα εργαλεία και τις γλώσσες προγραμματισμού της προτίμησής του, με σκοπό την δημιουργία εφαρμογών οι οποίες θα καλύπτουν τις ανάγκες του τελικού χρήστη. Οι επιλογές είναι πολλές ώστε ο προγραμματιστής να μπορέσει να εξασφαλίσει μια εφαρμογή που καλύπτει το χρήστη όσο αφορά τον έλεγχο του κτηρίου, την ενημέρωσή του για την κατάσταση στο κτήριο, αλλά παράλληλα να είναι φιλικές και η χρήση τους να είναι μια ευχάριστη εμπειρία. Τα είδη των εφαρμογών ποικίλουν από εφαρμογές προσωπικού υπολογιστή, διαδικτυακές εφαρμογές όπως και εφαρμογές έξυπνων συσκευών. Μια επιλογή είναι η χρήση λογισμικού με το οποίο ο χρήστης είναι ήδη εξοικειωμένος. Αυτό μπορεί να επιτευχθεί με ανάπτυξη πρόσθετου λογισμικού [24] το οποίο μπορεί να εγκαταστήσει ο χρήστης σε μια εφαρμογή την οποία ήδη χρησιμοποιεί. Υπάρχει ένας μεγάλος αριθμός εφαρμογών που χρησιμοποιεί ένας χρήστης στη καθημερινότητά του που έχουν την επιλογή επέκτασης μέσω πρόσθετου λογισμικού. Ουσιαστικά ο προγραμματιστής μπορεί να αναπτύξει εφαρμογές οι οποίες να εκμεταλλεύονται τις υπηρεσίες που προσφέρει ένα λογισμικό σε συνδυασμό με τις υπηρεσίες που παρέχει ένα σύστημα όπως το Überdust.

1.2 Υπάρχοντα Συστήματα και το σύστημα Überdust

Λόγω του τεράστιου ενδιαφέροντος για τα συστήματα ολοκληρωμένου ελέγχου, το επακόλουθο ήταν να υπάρξουν υλοποιήσεις συσκευών και λογισμικού που να κάνουν την δημιουργία και εγκατάσταση ενός συστήματος ολοκληρωμένου ελέγχου εφικτή. Η μεγαλύτερη πρόοδος έγινε στη δημιουργία συσκευών οι οποίες “κουμπώνουν” στις ηλεκτρικές συσκευές ενός σπιτιού όπως μια μηχανή καφέ, ένα κλιματιστικό ή τα φώτα ενός σπιτιού ώστε ο χρήστης μπορεί να τα ελέγξει από τον υπολογιστή του. Εταιρίες όπως η ELAN Home Systems ή Mile High Automation προσφέρουν υπηρεσίες για την εγκατάσταση των διαφόρων συσκευών εξασφαλίζοντας στον τελικό χρήστη την δυνατότητα του ελέγχου ενός κτηρίου.

Όμως οι υπηρεσίες αυτές είναι περιορισμένες στις δυνατότητες, κοστίζουν και δεν υπάρχει δυνατότητα παραμετροποίησης τους και επέκτασής τους καθώς χρησιμοποιείται κλειστό λογισμικό αλλά και κλειστό υλικό (hardware). Σε αντίθεση με αυτές της υπηρεσίες υπάρχει ανάπτυξη ολοκληρωμένων συστημάτων ελέγχου κτηρίων βασισμένη σε ανοιχτό λογισμικό και υλικό. Χαρακτηριστικό παράδειγμα η πλατφόρμα jHome η οποία είναι βασισμένη σε λογισμικό ανοιχτού κώδικα όπως η Java και το λειτουργικό σύστημα Android και σε μικροελεγκτές ανοιχτού υλικού όπως το arduino (σχήμα 1.2). Το jHome όχι μόνο έχει λιγότερο κόστος, λόγω του ότι είναι δωρεάν αλλά δίνει και την δυνατότητα σε ένα προγραμματιστή λογισμικού να το επεκτείνει ώστε να καλύψει τις ανάγκες του. Επίσης παρέχει διεπαφές λογισμικού (api) ώστε να μπορεί ο χρήστης να παραμετροποιήσει συγκεκριμένες λειτουργίες.



Σχήμα 1.2: Ο δημοφιλής μικροελεγκτής ανοιχτού υλικού arduino

1.2.1 Το Σύστημα Überdust

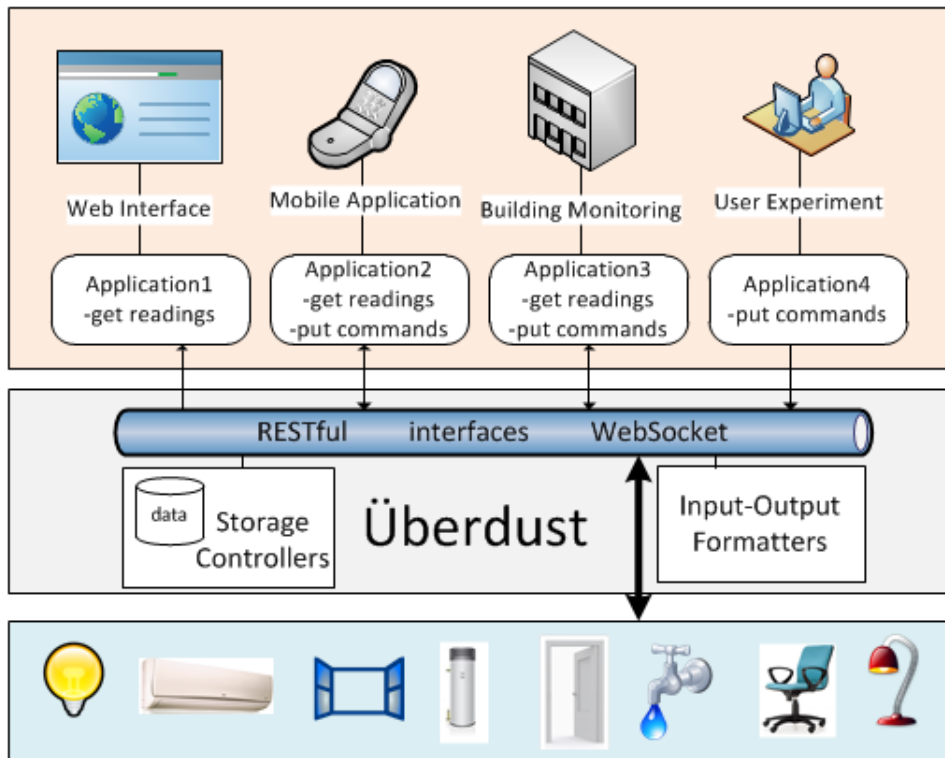
Ένα συστήματα ελέγχου κτηρίων αποτελείται από ένα επίπεδο ασύρματου δικτύου αισθητήρων, ένα υπολογιστικό σύστημα το οποίο επεξεργάζεται, διαχειρίζεται ή αποθηκεύει τα δεδομένα του δικτύου και τέλος ένα επίπεδο εφαρμογών μέσω του οποίου αλληλεπιδρά ο χρήστης.

Το Überdust [34] είναι μια αποθήκη δεδομένων και μια ενδιάμεση διαδικτυακή υπηρεσία η οποία συνδέει παρόχους και καταναλωτές δεδομένων όπως οι προγραμματιστές εφαρμογών, παρέχοντας δεδομένα πραγματικού χρόνου και ιστορικά δεδομένα. Ο κύριος στόχος του είναι να παρέχει χώρο για αποθήκευση δεδομένων, διαμοιρασμό και ανακάλυψη δεδομένων πραγματικού χρόνου.

Το Überdust είναι το νευρικό κέντρο του συστήματος καθώς είναι υπεύθυνο για την επικοινωνία των συσκευών του χαμηλού επιπέδου με τις εφαρμογές για τον έλεγχο του κτηρίου. Το πιο καινοτόμο χαρακτηριστικό του Überdust είναι ότι δεν επικεντρώνεται μόνο στο να συλλέγει δεδομένα αλλά και στο να επιτρέπει την επικοινωνία δύο κατευθύνσεων (bidirectional) μεταξύ των εφαρμογών και των συσκευών του ασύρματου δικτύου αισθητήρων. Το Überdust παρέχει ένα κεντριοποιημένο μηχανισμό για όλες τις συσκευές όσο αφορά την αποθήκευση ιστορικών δεδομένων, καθώς οι συσκευές λόγω των περιορισμών του υλικού τους, αδυνατούν να αποθηκεύσουν όλες τις απαραίτητες πληροφορίες στη μνήμη τους. Όλα τα μηνύματα από και προς τις συσκευές διέρχονται από το Überdust το οποίο σε γενικές γραμμές παρέχει τη λειτουργία ενός σημείου αναφοράς ώστε κάθε συσκευή να μπορεί να επικοινωνήσει με τις εφαρμογές και το αντίστροφο.

Πάνω από το Überdust είναι το επίπεδο των εφαρμογών οι οποίες ελέγχουν τα δεδομένα που έχουν συλλέξει οι αισθητήρες όπως και τους μηχανισμούς που αυτό προσφέρει. Οι περισσότεροι χρήστες που θέλουν να αλληλεπιδρούν απευθείας με τις συσκευές αισθητήρων το κάνουν σύμφωνα με το αφηρημένο σχέδιο που παρέχεται από αυτό το επίπεδο. Αυτό το αφηρημένο σχέδιο είναι από τα πιο ενδιαφέροντα περιεχόμενα του Überdust, καθώς οι συσκευές μπορούν να οργανωθούν σε ομάδες με βάση τις σημασιολογικές περιγραφές τους.

Το προαναφερθέν αφηρημένο σχέδιο ουσιαστικά είναι μια ομαδοποίηση αληθινών συσκευών σε μία κοινή σημασιολογική οντότητα. Σε σενάρια γραφειακών χώρων ή έξυπνων κτηρίων οι σημασιολογικές οντότητες καλύπτουν μικρές περιοχές όπως γραφεία, εργαστήρια, δωμάτια ορόφους. Αυτές οι σημασιολογικές οντότητες είναι πιο κατάλληλες για τον χρήστη από τις συσκευές καθεαυτές, καθώς προβάλλουν με ένα πιο φυσικό τρόπο τις δυνατότητες που έχουν οι συσκευές. Το Überdust από μόνο του παρέχει μία πολύ βασική διεπαφή, χωρίς μηχανισμό πιστοποίησης χρήστη για την δημοσίευση, πρόσβαση και αποστολή πληροφορίας στο σύστημα, λόγω του ότι είναι σχεδιασμένο ως ένα μηχανή προς



Σχήμα 1.3: Αρχιτεκτονική συστήματος ελέγχου κτηρίων με χρήση Überdust

μηχανή σύστημα (M2M [20]). Το Überdust προσφέρει μία διεπαφή Restull [26] και μία διεπαφή WebSocket [36].

Η διεπαφή Restfull είναι βασισμένη στο μοντέλο μεταφοράς αντιπροσωπευτικής κατάστασης υπηρεσιών ιστών (REST- REpresentational State Transfer). Παρέχει ένα εύκολο σε χρήση, πληροφοριακά προσδιορισμένο μοντέλο ώστε να εκθέσει τις υπάρχουσες υπηρεσίες. Η δημοσίευση και η κατανάλωση λειτουργιών υλοποιούνται χρησιμοποιώντας τις μεθόδους Get και Post του Πρωτόκολλου Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP).

Η διεπαφή Websocket προσφέρει τις ίδιες δυνατότητες χρησιμοποιώντας το IETF(Internet Engineering Task Force) τυποποιημένο WebSocket πρωτόκολλο. Το πρωτόκολλο Websocket είναι μια τεχνολογία που παρέχει διπλά κανάλια πλήρους επικοινωνίας δύο κατευθύνσεων πάνω από μία σύνδεση Πρωτοκόλλου Ελέγχου Μετάδοσης(TCP) η οποία μπορεί να χρησιμοποιηθεί από κάθε πελάτη, εφαρμογή ή διακομιστή. Χρησιμοποιώντας μια σύνδεση Websocket το πρόγραμμα μπορεί να λάβει καινούργια ή ιστορικά δεδομένα ή να δώσει εντολές για διάφορες ενέργειες. Επίσης οι συσκευές ελεγκτών μπορούν να χρησιμοποιήσουν μια WebSocket σύνδεση για να στείλουν καινούργιες πληροφορίες στη

διαδικτυακή υπηρεσία και να παραλάβουν αιτήσεις που να προωθήσουν στους αισθητήρες.

1.3 Στόχοι Διπλωματικής Εργασίας

Ο σκοπός αυτής της διπλωματικής είναι η δημιουργία ενός λογισμικού το οποίο να αλληλεπιδρά με ένα ήδη υπάρχον σύστημα αυτομάτου ελέγχου το οποίο ακολουθεί τα ανοιχτά πρότυπα και ο τελικός χρήστης να εκμεταλλεύεται τις δυνατότητές του μέσω του διαδικτύου. Οι στόχοι του λογισμικού αυτού είναι το τελικό αποτέλεσμα να καλύπτει τις ανάγκες του χρήστη και να είναι φιλικό ως προς την χρήση. Ο τρόπος χρήσης του λογισμικού θα πρέπει να είναι γνώριμος και ο χρήστης να έρχεται όσο το δυνατόν λιγότερο σε επαφή με τεχνικές λεπτομέρειες.

Επίσης λόγω του ότι οι ανάγκες αυξάνονται όπως και οι δυνατότητες ενός συστήματος ελέγχου κτηρίων, το λογισμικό θα πρέπει να μπορεί να επεκταθεί ώστε να προστεθούν επιπλέον δυνατότητες. Ο υπεύθυνος για την διαχείριση και την ανάπτυξη του συστήματος ελέγχου θα πρέπει να έχει την δυνατότητα να κάνει τις απαραίτητες αλλαγές στο λογισμικό ώστε αυτό να εξασφαλίσει την ομαλή του λειτουργία.

Το λογισμικό SkypeDust [28], δηλαδή το αντικείμενο της διπλωματικής θα βασιστεί στις λειτουργίες του Skype [27] μια εφαρμογή ομιλίας μέσω διαδικτύου αλλά και αποστολής μηνυμάτων. Το Überdust είναι το σύστημα αυτομάτου ελέγχου ανοιχτού λογισμικού το οποίο θα χρησιμοποιηθεί.

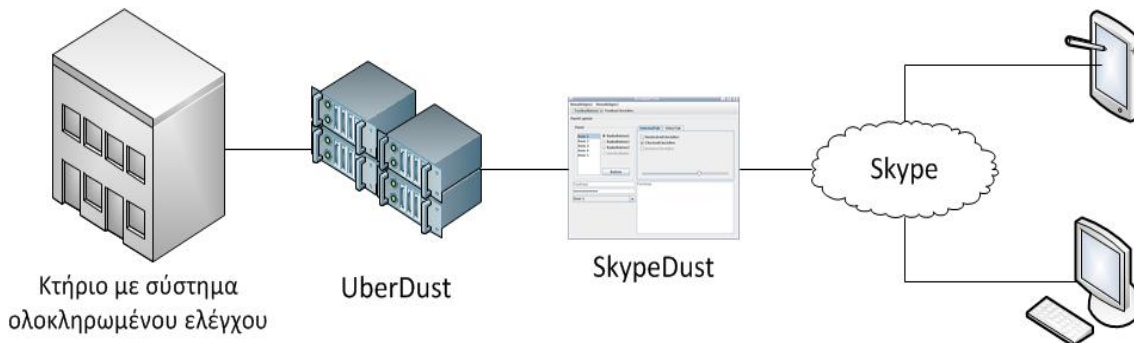
1.4 Συνεισφορά Διπλωματικής Εργασίας

Στα πλαίσια της διπλωματικής εργασίας αναπτύχθηκε το λογισμικό SkypeDust το οποίο λειτουργεί ως ένα ενδιάμεσο λογισμικό μεταξύ των υπηρεσιών του Überdust και των υπηρεσιών του Skype, διασφαλίζοντας την εύχρηστη επικοινωνία του χρήστη με ένα σύστημα ολοκληρωμένου ελέγχου κτηρίου όπως το Überdust (σχήμα 1.4).

Το SkypeDust αναλαμβάνει να προωθεί τα μηνύματα του χρήστη μέσω της εφαρμογής Skype στο Überdust, όπως και να προωθεί τα δεδομένα που στέλνει ο Überdust στον χρήστη μέσω μηνυμάτων της εφαρμογής Skype.

Το αίτημα που στέλνει ο χρήστης μέσω Skype στο Überdust τα λαμβάνει ο SkypeDust τα επεξεργάζεται κατάλληλα και προωθεί τα αιτήματα στο Überdust. Αργότερα όταν το Überdust θα θελήσει να στείλει μια ειδοποίηση ή δεδομένα

στον χρήστη ο SkypeDust θα αναλάβει να πάρει αυτά τα δεδομένα να τα επεξεργαστεί και να τα προωθήσει στον χρήστη μέσω Skype.



Σχήμα 1.4: Γενική αρχιτεκτονική του συστήματος

Χάρης σε αυτήν την επικοινωνία το SkypeDust παρέχει την υπηρεσία άμεσης ενημέρωσης για την κατάσταση του χώρου ενός κτηρίου, πιο συγκεκριμένα ο χρήστης μπορεί να ελέγξει τον φωτισμό, την θερμοκρασία, την υγρασία, τον θόρυβο αλλά και την παρουσία ατόμων σε ένα δωμάτιο.

Επιπλέον παρέχει την υπηρεσία ελέγχου των συσκευών που είναι κατανεμημένες στο κτήριο ώστε να μπορεί ο χρήστης να μεταβάλει την κατάσταση στους χώρους ενός κτηρίου. Επίσης δίνεται η υπηρεσία προειδοποίησης για τις αλλαγές που συμβαίνουν σε ένα χώρο αφού ο χρήστης έχει απαιτήσει την συνεχή ενημέρωσή του.

Ακόμη το SkypeDust προσφέρει την δυνατότητα ελέγχου της κατάστασης των συσκευών-κόμβων που είναι κατανεμημένες στο σύστημα ώστε να αποφευχθούν πιθανές περιπτώσεις βλάβης.

Πέρα από τις υπηρεσίες που αφορούν το σύστημα του ολοκληρωμένου ελέγχου το SkypeDust παρέχει την δυνατότητα στον χρήστη να κάνει την χρήση του λογισμικού πιο εύκολη και γρήγορη. Ο χρήστης μπορεί να δώσει δικές του ονομασίες σε χαρακτηριστικά του κτηρίου αλλά και σε ενέργειες του συστήματος ελέγχου διατηρώντας έτσι ένα αφηρημένο επίπεδο μεταξύ του συστήματος ελέγχου και το SkypeDust.

Τέλος Το λογισμικό SkypeDust είναι ευέλικτο, ανοιχτού κώδικα και επεκτάσιμο. Συγκεκριμένα παρέχει την δυνατότητα σε έναν μηχανικό λογισμικού να το επεκτείνει με πρόσθετο λογισμικό που έχει αναπτύξει. Επιπλέον ο μηχανικός λογισμικού είναι σε θέση να πειραματιστεί πάνω στον πηγαίο κώδικα του SkypeDust και να καλύψει τις ανάγκες του.

1.5 Οργάνωση Διπλωματικής Εργασίας

Στο σημείο αυτό τελειώνει η εισαγωγή στα ολοκληρωμένα συστήματα ελέγχου κτηρίων και ή περιγραφή του συστήματος του οποίου αυτή η διπλωματική αποτελεί μέρος. Δόθηκε περιγραφή για τα επίπεδα του συστήματος δηλαδή του δικτύου αισθητήρων, τον διακομιστή αποθήκευσης δεδομένων *Überdust* και το λογισμικό το οποίο εξασφαλίζει την διαχείριση του συστήματος από το χρήστη.

Στο Δεύτερο κεφάλαιο περιγράφεται η χρήση των υπηρεσιών που παρέχει το λογισμικό *Skype* για την ανάπτυξη του λογισμικού που αλληλεπιδρά με το ολοκληρωμένο σύστημα. Αρχικά παρουσιάζονται οι λόγοι της επιλογής των υπηρεσιών του *Skype* και τι αντίκτυπο έχουν στη καθημερινή ζωή ενός χρήστη. Γίνεται αναφορά για τους τρόπους χρήσης των υπηρεσιών του και πως γίνεται ένας προγραμματιστής να τις χρησιμοποιήσει στο λογισμικό του. Παρουσιάζεται ο σκοπός της ανάπτυξης ενός λογισμικού με αυτά τα χαρακτηριστικά, οι προδιαγραφές που πρέπει να καλύψει και η συνεισφορά του στην αλληλεπίδραση ενός χρήστη με ένα σύστημα ολοκληρωμένου ελέγχου.

Στο τρίτο κεφάλαιο δίνεται μια γενική περιγραφή της εφαρμογής που αναπτύχθηκε. Γίνεται αναφορά όσο αφορά τη διεπαφή χρήστη, την επικοινωνία με το *Überdust*, τις ρυθμίσεις του λογισμικού, την αποθήκευση δεδομένων και την δυνατότητα εγκατάστασης πρόσθετου λογισμικού. Επίσης δίνεται παράδειγμα χρήσης της εφαρμογής.

Στο Τέταρτο κεφάλαιο παρατίθεται αναλυτική περιγραφή της αρχιτεκτονικής του λογισμικού και τα μέρη στα οποία χωρίζεται. Έπειτα γίνεται περιγραφή των μερών του λογισμικού, το στόχο τους και την υλοποίηση τους. Ποιες ήταν οι επιλογές όσο αφορά την υλοποίηση και τι εργαλεία χρησιμοποιήθηκαν. Τέλος γίνεται αναφορά για την συνολική λειτουργία της αρχιτεκτονικής και πως αλληλεπιδρούν τα διάφορα επίπεδα μεταξύ τους.

Στο πέμπτο κεφάλαιο παρουσιάζεται η λειτουργία του μηχανισμού πρόσθετου λογισμικού και ποιος είναι ο ρόλος του. Παρουσιάζονται οι επιλογές στην εγκατάσταση πρόσθετου λογισμικού και η λειτουργικότητα που αυτό μπορεί να δώσει. Παρουσιάζονται οι τρόποι με τους οποίους μπορεί να χρησιμοποιηθεί το πρόσθετο λογισμικό ώστε ο προγραμματιστής να μην χρειάζεται να αλλάξει τον πηγαίο κώδικα για να έχει την επιθυμητή λειτουργικότητα. Τέλος παρουσιάζονται οι προδιαγραφές που πρέπει να υποστηρίζει το κάθε κομμάτι λογισμικού ώστε να είναι συμβατό με την αρχιτεκτονική του συστήματος και να είναι δυνατή η εναλλαγή των κομματιών λογισμικού κατά την διάρκεια εκτέλεσης του προγράμματος.

Στο έκτο κεφάλαιο παρατίθενται τα συμπεράσματα της διπλωματικής εργασίας και γίνεται αναφορά σε μελλοντικούς στόχους .

Κεφάλαιο 2

Η χρήση υπηρεσιών Skype

Στο κεφάλαιο αυτό παρουσιάζονται οι υπηρεσίες που προσφέρει το Skype σε έναν μηχανικό λογισμικού και πως χρησιμοποιούνται στην εφαρμογή. Θα αναφερθούμε στο Skype και τους λόγους που το επιλέξαμε, τους τρόπους με του οποίους μας δίνει την δυνατότητα να χρησιμοποιήσουμε τις υπηρεσίες του στην εφαρμογή μας, και στο πως συνδυάζουμε σε αυτήν την εφαρμογή τις υπηρεσίες του Skype και του Überdust.

2.1 Το λογισμικό Skype

Το Skype [27] είναι μια εφαρμογή λογισμικού η οποία δημιουργήθηκε το 2003 και πλέον ανήκει στην εταιρεία Microsoft . Η υπηρεσία επιτρέπει στους χρήστες να επικοινωνήσουν είτε τηλεφωνικά μέσω διαδικτύου (voip [35]), είτε με βιντεοκλήση ή άμεση αποστολή μηνυμάτων μέσω internet. Επίσης οι χρήστες μπορούν να επικοινωνήσουν υλοποιώντας κλήσεις σε αριθμούς σταθερών και κινητών τηλεφώνων. Επιπλέον το Skype δίνει δυνατότητα και σε άλλες λειτουργίες όπως τη μεταφορά δεδομένων.

Ένας χρήστης μπορεί να χρησιμοποιήσει μια εφαρμογή Skype με διάφορες συσκευές. Μπορεί να την εγκαταστήσει στον προσωπικό του υπολογιστή ώστε να έχει όλες τις λειτουργίες που παρέχει το Skype ,ή να εγκαταστήσει σε έναν υπολογιστή ταμπλέτα ή σε ένα έξυπνο τηλέφωνο το αντίστοιχο λογισμικό με τις ίδιες ή λιγότερες λειτουργίες.

Ουσιαστικά πρόκειται για μια εφαρμογή όπου ο χρήστης μπορεί να έχει εύκολα και άμεσα πρόσβαση. Η μόνη προϋπόθεση για τη χρήση της εφαρμογής είναι μία συσκευή για την οποία υπάρχει διαθέσιμο το αντίστοιχο Skype λογισμικό.

Το Skype έχει 663 εκατομμύρια χρήστες οι οποίοι ολοένα αυξάνονται. Οι πι-



Σχήμα 2.1: Το λογότυπο του Skype

Θαυμάσιες είναι οι δυνατότητες ενός χρήστη έξυπνου κινητού ή προσωπικού υπολογιστή να είναι εξοικειωμένος με το λογισμικό Skype είναι μεγάλες. Λόγω του ότι ένας χρήστης μπορεί να είναι ήδη εξοικειωμένος με την χρήση του Skype, μια εφαρμογή που αφορά τις υπηρεσίες του Skype έχει αυξημένες πιθανότητες ο τρόπος λειτουργίας της να είναι γνώριμος στον χρήστη και αυτός να αρχίσει να τη χρησιμοποιεί απευθείας.

Ένα από τα μεγαλύτερα πλεονεκτήματα του Skype είναι η συνέπεια όσο αφορά τις υπηρεσίες επικοινωνίας. Μπορεί να δουλέψει με την πλειονότητα των τοίχων προστασίας (firewalls) και πύλες δικτύων (gateways) χωρίς ο χρήστης να μπει στη διαδικασία για ειδικές ρυθμίσεις. Αυτό το γεγονός συμβάλλει στην ευκολία χρήσης του Skype από χρήστες οι οποίοι δεν έχουν τεχνολογικό υπόβαθρο.

Το Skype είναι ασφαλές όσο αφορά τα δεδομένα που στέλνονται μεταξύ δύο χρηστών. Για τα δεδομένα που στέλνονται χρησιμοποιούνται αλγόριθμοι παγκόσμια αναγνωρισμένοι. Αυτοί οι αλγόριθμοι προστατεύουν την επικοινωνία από το να πέσουν στα χέρια ηλεκτρονικών εγκληματιών. Με αυτόν τον τρόπο το Skype διασφαλίζει τα προσωπικά δεδομένα όπως και την ακεραιότητα των δεδομένων που στέλνονται μεταξύ των λογαριασμών.

Λόγω της μεγάλης ζήτησης εφαρμογών σε υπηρεσίες γραπτών μηνυμάτων, διαδικτυακής συνομιλίας και ηχητικού ηλεκτρονικού ταχυδρομείου (voicemail) το Skype δίνει την δυνατότητα στους προγραμματιστές για ανάπτυξη εφαρμογών που μπορούν να χρησιμοποιήσουν της αντίστοιχες υπηρεσίες του Skype ώστε το τελικό προϊόν να παρέχει τις επιθυμητές λειτουργίες.

2.2 Εισαγωγή στη προγραμματιστική χρήση υπηρεσιών Skype

Υπάρχουν δύο επιλογές για την ανάπτυξη μια εφαρμογής που χρησιμοποιεί τις υπηρεσίες του Skype. Η πρώτη είναι με την χρήση του desktop api (Διεπαφή

Προγραμματισμού Εφαρμογών προσωπικού υπολογιστή). Πιο συγκεκριμένα ο χρήστης πρέπει να έχει το λογισμικό Skype εγκατεστημένο στον προσωπικό του υπολογιστή και μια εφαρμογή να αλληλεπιδρά με την εφαρμογή Skype μέσω IPC [16] (διαδικρασιακή επικοινωνία). Το desktop api πρωτόκολλο βασίζεται στην επικοινωνία με την desktop εφαρμογή του Skype μεταφέροντας δεδομένα μέσω IPC υπό μορφή εντολών οι οποίες μεταφράζονται από το λογισμικό Skype σε αντίστοιχες λειτουργίες. Η δεύτερη επιλογή είναι η ανάπτυξη λογισμικού με την χρήση του SkypeKit api [29].

Το SkypeKit είναι μια συλλογή από προγραμματιστικές διεπαφές εφαρμογών (Api [4]) τα οποία επιτρέπουν σε συσκευές ή λογισμικό συνδεδεμένο στο διαδίκτυο, να προσφέρουν τις υπηρεσίες του Skype. Το λογισμικό είναι σχεδιασμένο να δουλεύει με ένα εύρος ολοκληρωμένων κυκλωμάτων (chipsets), λειτουργικά συστήματα και συσκευές ήχου-βίντεο. Ο προγραμματιστής χρησιμοποιεί το Skype runtime, ένα εκτελέσιμο αρχείο το οποίο είναι ουσιαστικά μια εφαρμογή Skype χωρίς γραφικό περιβάλλον για το χρήστη. Το εκτελέσιμο αρχείο λαμβάνει εντολές για τις υπηρεσίες που αφορούν το Skype μέσω μεθόδων IPC. Χάρη στο Skypekit api ο προγραμματιστής έχει βιβλιοθήκες και εργαλεία για την επικοινωνία με το Skype εκτελέσιμο μέσω IPC και επομένως μέσα στο το πρόγραμμα του μπορεί να προσθέσει κώδικα ώστε να δίνει εντολές στο εκτελέσιμο αρχείο Skype.

Το Skypekit παρέχει ένα σύνολο από μεθόδους και κλάσεις [7] που δίνουν τις δυνατότητες στον προγραμματιστή ώστε να εισέλθει στο σύστημα να βγει και να διαχειριστεί λογαριασμούς Skype. Μπορεί να προσθέσει και να διαχειριστεί λογαριασμούς επαφών όπως και να δημιουργήσει ομάδες. Επίσης υπάρχει η δυνατότητα λήψης και αποστολής ασύγχρονων μηνυμάτων όπως και η δυνατότητα έναρξης συνομιλιών. Επιπλέον δίνεται η δυνατότητα μεταφοράς αρχείων οποιουδήποτε μεγέθους κατά την διάρκεια συνομιλίας.

Αρχείο python 2.1: Πρόγραμμα python που πραγματοποιεί εισόδο στο Skype και συνομιλία μέσω SkypeKit

```
import sys
import keypair, Skype
from time import sleep

loggedIn = False

def AccountOnChange (self, property_name):
    global loggedIn
    if property_name == 'status':
        print ('Login sequence: ' + self.status);
        if self.status == 'LOGGED_IN':
            loggedIn = True
        if self.status == 'LOGGED_OUT':
            loggedIn = False

def OnMessage(self,
```

```
message, changesInboxTimestamp, supersedesHistoryMessage, conversation):
    if message.author != accountName:
        print(message.author_displayname + ': ' + message.body_xml);
        conversation.PostText('Automated reply Yo.', False);

if __name__ == '__main__':

    accountName = "yourusername"
    accoutPsw = "yourpassword"

    try:
        MySkype = Skype.GetSkype(keypair.keyFileName)
    except: print("no skype instanse")

    Skype.Account.OnPropertyChange = AccountOnChange

    myaccount = MySkype.GetAccount(accountName)
    myaccount.LoginWithPassword(accoutPsw, False, False)

    while not loggedIn:
        sleep(1)

    skypeContactGroup = MySkype.GetHardwiredContactGroup("ALL_KNOWN_CONTACTS")
    skypeContacts = skypeContactGroup.GetContacts();

    for scontact in skypeContacts: print(scontact.displayname)

    userchat = str(raw_input("choose a user to chat; "))

    conversation = MySkype.GetConversationByIdentity(userchat)

    while True:
        sendtext = str(raw_input("send text: "))
        conversation.PostText(sendtext, False)

    MySkype.stop()
```

Οι δυνατότητες που μας δίνονται είναι τεράστιες ώστε να δημιουργηθεί λογισμικό το οποίο να τηρεί τις προϋποθέσεις που απαιτεί ένα σύστημα κρίσιμης σημασίας όπως ένα σύστημα ολοκληρωμένου ελέγχου κτηρίου. Εξασφαλίζουμε την ασφάλεια στη μεταφορά δεδομένων, την εγκυρότητα τους, την αμεσότητα λόγω της ασύγχρονης επικοινωνίας και την ευκολία στη χρήση λόγω της εξοικείωσης του χρήστη με το λογισμικό Skype.

Κεφάλαιο 3

Η εφαρμογή SkypeDust

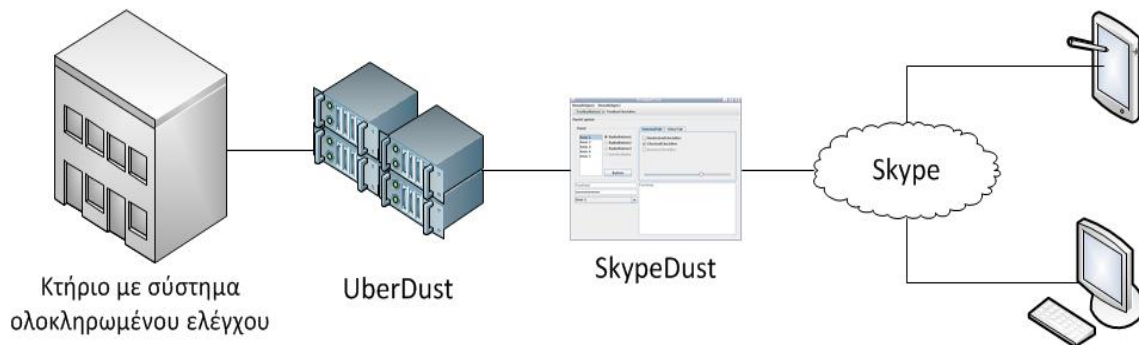
Στο κεφάλαιο αυτό δίνεται η περιγραφή της γενικής αρχιτεκτονικής του συστήματος SkypeDust. Έπειτα γίνεται παρουσίαση της εφαρμογής SkypeDust, των λειτουργιών και των δυνατοτήτων που παρέχει και τον τρόπο χρήσης τους από έναν άτομο. Επίσης περιγράφεται η προεπιλεγμένη εκτέλεση της εφαρμογής από έναν χρήστη όπως και η εκτέλεση της εφαρμογής με χρήση πρόσθετου λογισμικού

3.1 Γενική αρχιτεκτονική συστήματος SkypeDust

Το SkypeDust λειτουργεί ως ένα ενδιάμεσο λογισμικό το οποίο διασφαλίζει την επικοινωνία χρήστη-Überdust μέσω Skype με φιλικό τρόπο χωρίς ο χρήστης να έρθει αντιμέτωπος με τις ιδιαιτερότητες του Überdust. Το SkypeDust καταναλώνει δύο ειδών δεδομένα. Τα δεδομένα του Überdust που αφορά την πληροφορία που στέλνει και μπορεί να δώσει το Überdust στις διάφορες μορφές, και τα δεδομένα μέσω μηνυμάτων της υπηρεσίας Skype από τα οποία προκύπτουν οι αιτήσεις του χρήστη προς το σύστημα κτηρίου που είναι εγκατεστημένο. Ο άλλος ρόλος που έχει το SkypeDust είναι η αποστολή αιτημάτων τα οποία θα καθορίσουν τις ενέργειες που θα λάβει το σύστημα. Τέλος αναλαμβάνει να παρουσιάσει τα δεδομένα που λαμβάνει από το Überdust στον χρήστη, με φιλικό, άμεσο και γρήγορο τρόπο χρησιμοποιώντας την υπηρεσία μηνυμάτων του Skype.

Παρακάτω (σχήμα 3.1) βλέπουμε τη συνδυασμένη λειτουργία ενός κτηρίου με δίκτυο αισθητήρων, του Überdust, της εφαρμογής SkypeDust και ενός υπολογιστή ή φορητής ηλεκτρονικής συσκευής.

Ο απλός χρήστης μέσω ενός μηνύματος στο Skype δίνει μια εντολή στη Skype επαφή η οποία είναι συνδεδεμένη στο SkypeDust. Η εντολή αφορά την ενημέ-



Σχήμα 3.1: Γενική αρχιτεκτονική του συστήματος

ρωση του χρήστη για τον φωτισμό σε ένα συγκεκριμένο δωμάτιο του κτηρίου. Το μήνυμα θα σταλεί στον αντίστοιχο λογαριασμό που διαχειρίζεται το SkypeDust. Το SkypeDust θα επεξεργαστεί το μήνυμα και θα δημιουργήσει το αντίστοιχο αίτημα το οποίο θα σταλεί στο Überdust. Το Überdust ως διακομιστής (server) αποθήκευσης δεδομένων πραγματικού χρόνου συλλέγει τα δεδομένα του δικτύου αισθητήρων που είναι εγκατεστημένο στο κτήριο. Με το που σταλεί το αίτημα από το SkypeDust μέσω του Restfull api του Überdust, ο Überdust θα απαντήσει στο αίτημα του SkypeDust. Αργότερα τα δεδομένα θα τα επεξεργαστεί το SkypeDust και θα στείλει μήνυμα μέσω του Skype στον χρήστη. Ο χρήστης θα έχει ενημερωθεί για την κατάσταση του φωτισμού στο δωμάτιο.

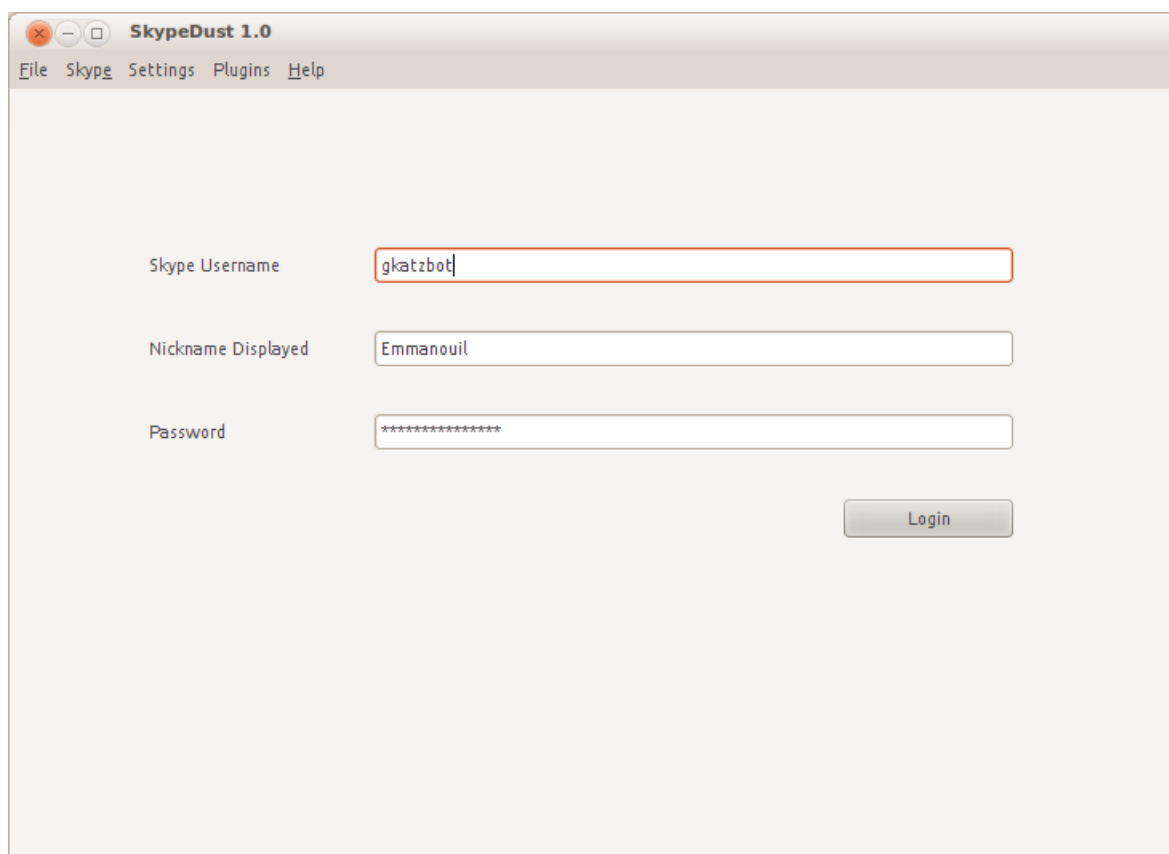
3.2 Περιγραφή της εφαρμογής

3.2.1 Διαχείριση Λογαριασμού

Το SkypeDust είναι μια εφαρμογή Java [18] προσωπικού υπολογιστή με γραφικό περιβάλλον. Η επιλογή για το γραφικό περιβάλλον ήταν η εργαλειοθήκη Swing [32] της Java. Αρχικά ο χρήστης για να μπορέσει να χρησιμοποιήσει την εφαρμογή πρέπει να εισάγει έναν έγκυρο λογαριασμό Skype με τα στοιχεία όπως το όνομα χρήστη (username), των κωδικό (password) όπως και το ψευδώνυμο που θα εμφανίζεται στις επαφές του λογαριασμού (σχήμα 3.2).

Ο χρήστης μπορεί να εισέλθει στο σύστημα με ένα μόνο λογαριασμό. Λόγω των περιορισμών του SkypeKit, για να μπορέσει να εισέλθει με ένα διαφορετικό λογαριασμό από αυτόν που έχει ήδη εισέλθει πρέπει να εισάγει τον καινούργιο λογαριασμό και να ξανανοίξει την εφαρμογή. Τα στοιχεία του χρήστη αποθηκεύονται στον δίσκο ώστε να μη χρειάζεται να εισάγει ξανά τα στοιχεία του.

Από τις επαφές που έχει ο λογαριασμός που εισάγει ο χρήστης στο σύστημα,



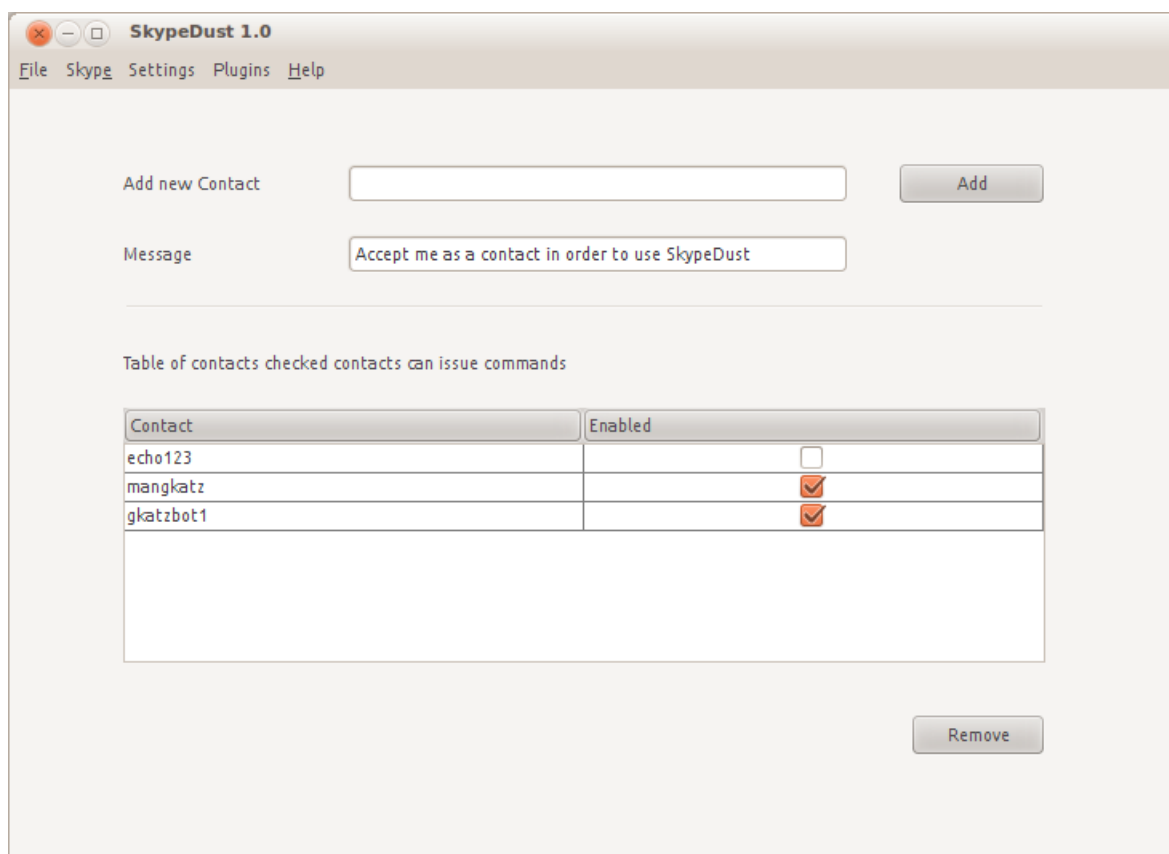
Σχήμα 3.2: Οθόνη ρυθμίσεων λογαριασμού.

μόνο ορισμένες από αυτές θα μπορούν να έχουν τη δυνατότητα να δίνουν εντολές μέσω Skype. Η εφαρμογή δίνει τη δυνατότητα στον χρήστη να επιλέξει ποιες από τις αυτές τις επαφές θα μπορούν να έχουν αυτή τη δυνατότητα. Από την εφαρμογή ο χρήστης μπορεί να προσθέσει καινούργια επαφή στον λογαριασμό που είναι αποθηκευμένος στο σύστημα. Επίσης δίνεται η δυνατότητα να διαγράψει κάποιες από τις επαφές που δεν χρειάζεται (σχήμα 3.3).

Το αποτέλεσμα είναι η εφαρμογή να είναι ανεξάρτητη από το επίσημο λογισμικό Skype κάνοντάς την πιο απλή για το χρήστη. Προφανώς όμως ο χρήστης μπορεί να κάνει τις αντίστοιχες ρυθμίσεις για τον λογαριασμό από την εφαρμογή Skype.

3.2.2 Ονοματολογία κόμβων και δυνατοτήτων

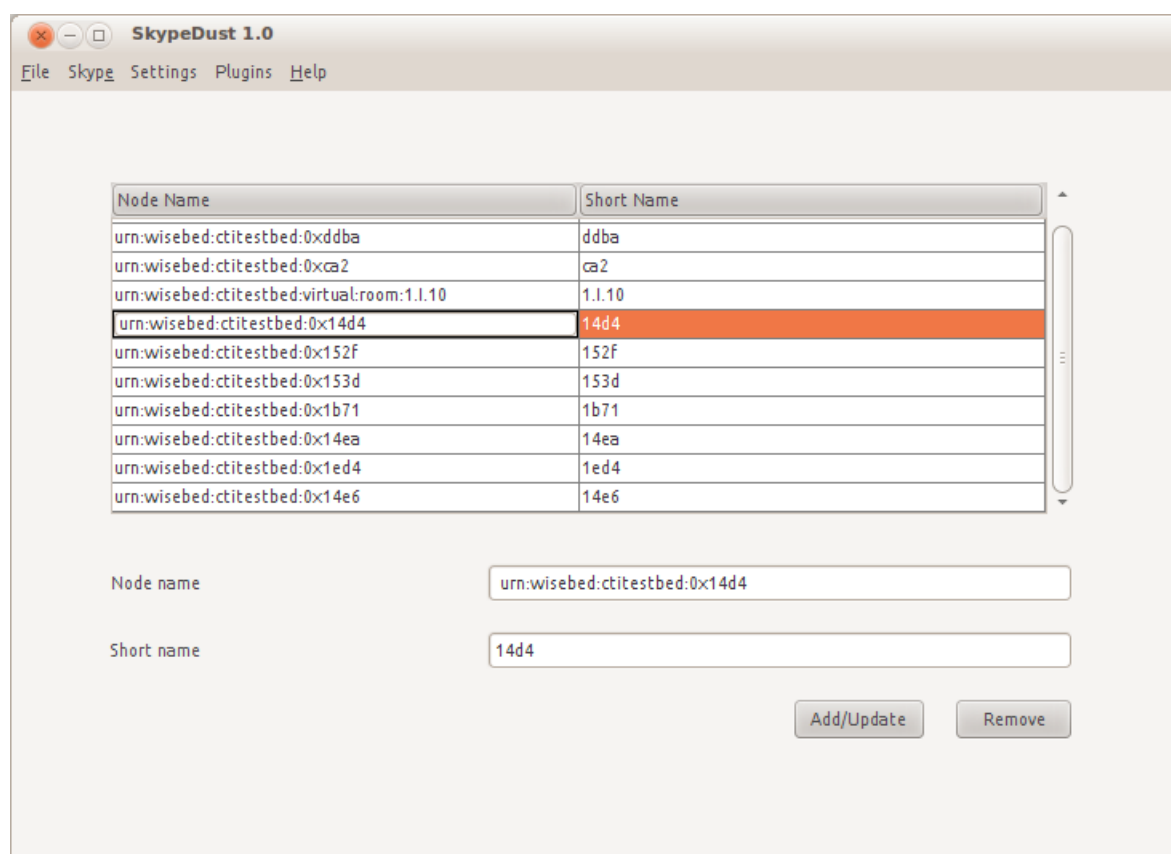
Το Überdust έχει μια σειρά από κόμβους. Αυτοί οι κόμβοι είναι μικροελεγκτές με αισθητήρες κατανεμημένους στο σύστημα με διαφορετικές ονομασίες.



Σχήμα 3.3: Ρύθμιση επαφών.

Λόγω του αριθμού τους η ονομασία είναι περίπλοκη και χρειάζεται ένας μεγάλος αριθμός χαρακτήρων για την αναπαράστασή τους. Το ίδιο ισχύει και για τις δυνατότητες που έχει ένας κόμβος. Λόγω του ότι το σύστημα έχει μια πληθώρα από ιδιότητες και ενέργειες που παρέχει η ονοματολογία που προκύπτει είναι μεγάλη σε χαρακτήρες και συνήθως προκύπτει από τα χαρακτηριστικά του συστήματος. Ο χρήστης για να μπορέσει μέσα από το Skype να ελέγξει μια ιδιότητα ενός κόμβου θα πρέπει να θυμάται το όνομα της ιδιότητας αλλά και το όνομα του κόμβου. Ακόμα και αν θυμάται αυτά τα δύο στοιχεία θα πρέπει να πληκτρολογήσει στην εφαρμογή Skype πολλούς χαρακτήρες ώστε να σταλεί το αίτημα του στο Überdust και να γίνει η αντίστοιχη ενέργεια. Έτσι η εφαρμογή δεν είναι φιλική προς το χρήστη ιδικά σε περιπτώσεις έξυπνων κινητών τηλεφώνων όπου η πληκτρολόγηση χαρακτήρων είναι χρονοβόρα και δύσκολη.

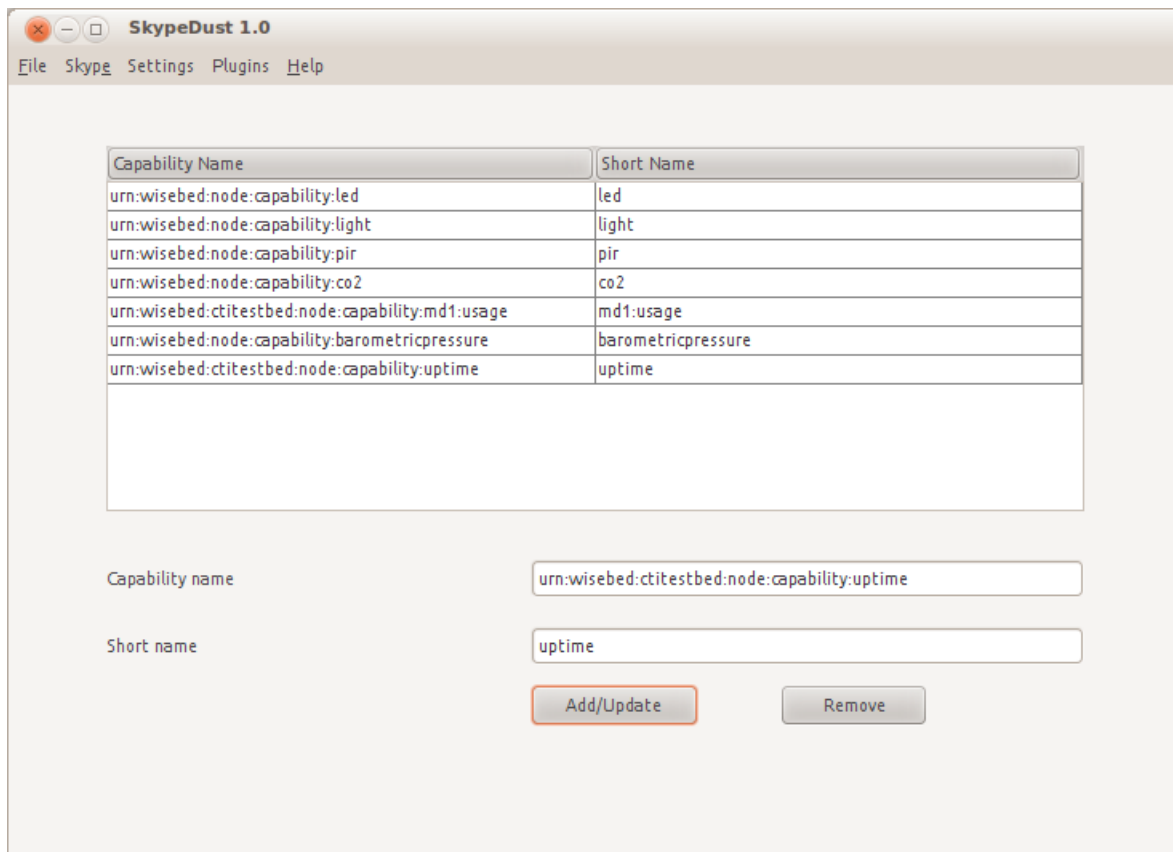
Αυτό το πρόβλημα η εφαρμογή SkypeDust το λύνει δίνοντας στον χρήστη την επιλογή να δώσει μια σύντομη ονομασία για τους κόμβους αλλά και τις δυνατότητες. Στην οθόνη επιλογής κόμβου ο χρήστης μπορεί να εισάγει έναν



Σχήμα 3.4: Οθόνη σύντομης ονομασίας κόμβου.

καινούργιο κόμβο από τον Überdust με το αντίστοιχο κωδικό όνομα. Επίσης δίνεται η δυνατότητα στον χρήστη να αλλάξει μια σύντομη ονομασία κόμβου που έχει ήδη δώσει (σχήμα 3.4).

Η οθόνη ονομασίας ιδιότητας έχει την ίδια λειτουργικότητα με την αντίστοιχη οθόνη κόμβου. Ο χρήστης εισάγει την ιδιότητα στην οποία θέλει να δώσει σύντομη ονομασία μαζί με την σύντομη ονομασία της. Επίσης ο χρήστης έχει την δυνατότητα να αλλάξει την ονομασία που έχει ήδη δώσει (σχήμα 3.5).

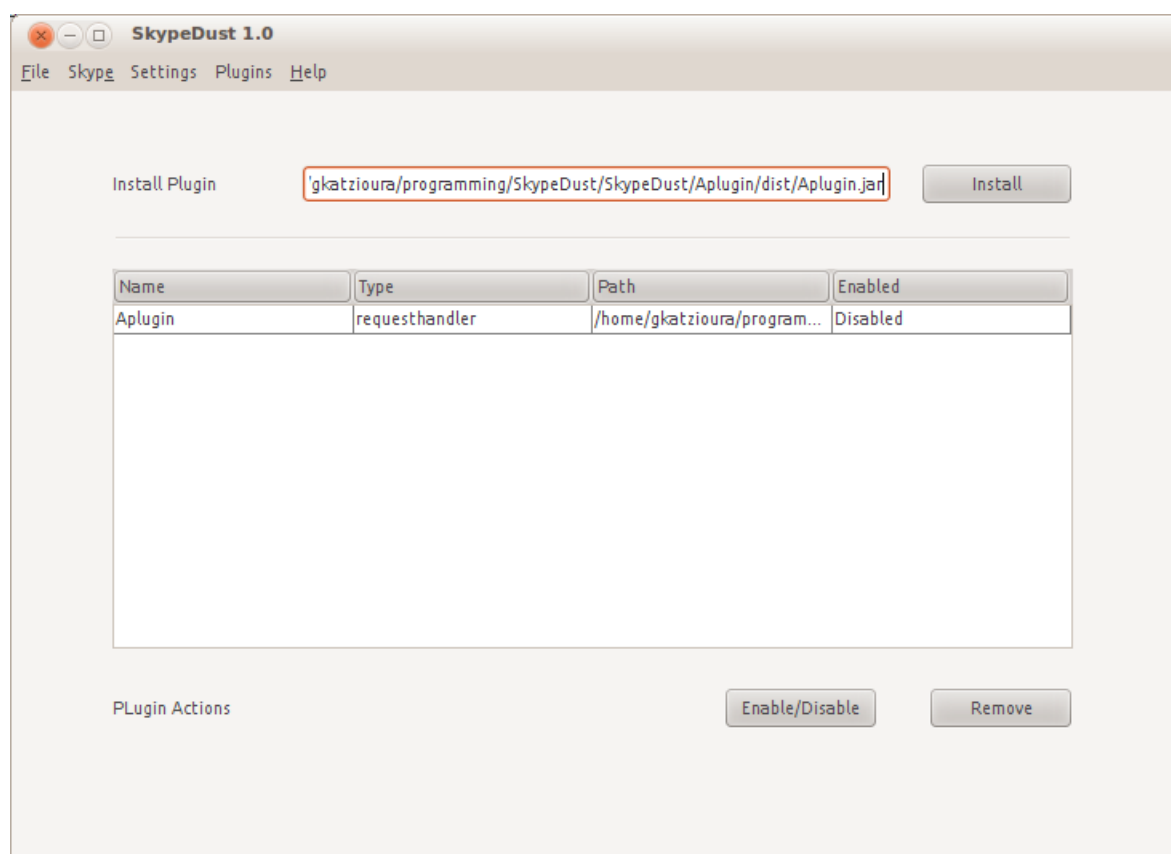


Σχήμα 3.5: Οθόνη σύντομης ονομασίας ιδιότητας.

3.2.3 Διαχείριση plugins

Η εφαρμογή για την εξυπηρέτηση του χρήστη έχει μια προκαθορισμένη λειτουργία στα αιτήματα που αυτός στέλνει. Οι ανάγκες ενός χρήστη μεταβάλλονται όπως και οι προτιμήσεις του όσο αφορά τις αιτήσεις και τις λειτουργίες που θέλει να εκτελεστούν στο Überdust. Επομένως οι πιθανότητες η προκαθορισμένη λειτουργία του SkypeDust να μην καλύπτει τις ανάγκες του χρήστη είναι πολλές. Για αυτόν τον λόγο η εφαρμογή δίνει τη δυνατότητα στον χρήστη να εγκαταστήσει πρόσθετο λογισμικό το οποίο μπορεί να αλλάξει τη λειτουργία του SkypeDust όσο αφορά τα αιτήματα που θα λάβει από το χρήστη, και τις αντίστοιχες αιτήσεις που θα στείλει στο Überdust (σχήμα 3.6).

Ο χρήστης μπορεί να εισάγει ένα συμπιεσμένο αρχείο Java(Jar [17]) από το σύστημα αρχείων του (filesystem). Αυτό το αρχείο για να γίνει δεκτό πρέπει να πληρεί κάποιες προϋποθέσεις. Μέσα το Jar αρχείο πρέπει να υπάρχει ένα αρχείο επεκτάσιμης γλώσσας σήμανσης (XML) με τις ρυθμίσεις του και τις αντίστοιχες κλάσεις. Ανάλογα με αυτά τα αρχεία η εφαρμογή θα ξέρει τι είδους πρόσθετο



Σχήμα 3.6: Διαχείριση πρόσθετου λογισμικού

λογισμικό θα εγκατασταθεί και που ακριβώς θα χρησιμοποιηθεί. Αφού το πρόσθετο λογισμικό εγκατασταθεί τότε ο χρήστης θα μπορεί να το ενεργοποιήσει ώστε να έχει τις επιθυμητές λειτουργίες.

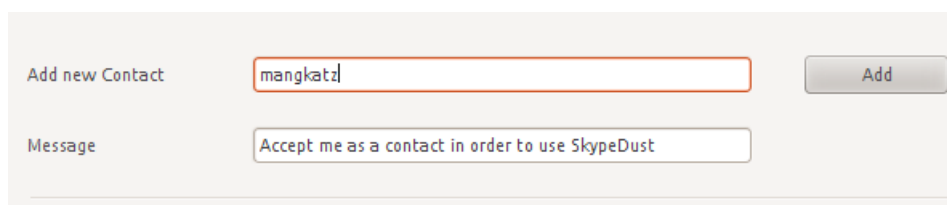
3.3 Εκτέλεση της εφαρμογής

3.3.1 Προεπιλεγμένη εκτέλεση εφαρμογής

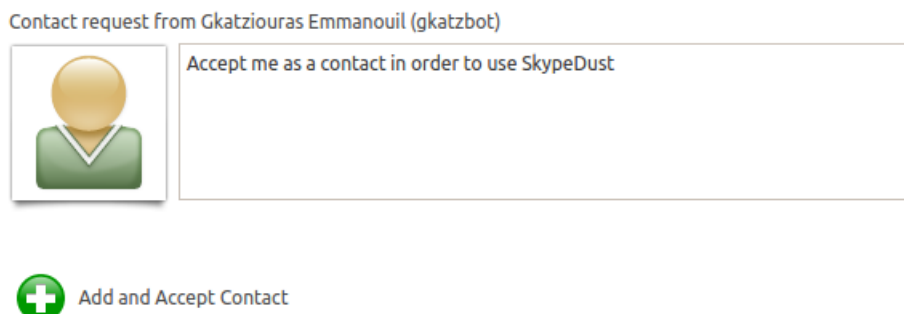
Σε αυτό το σημείο θα δοθεί μια αναφορά στην εκτέλεση της εφαρμογής με τις προεπιλεγμένες ρυθμίσεις και λειτουργίες. Δηλαδή τις λειτουργίες που η εφαρμογή είναι ρυθμισμένη να παρέχει όταν ο διαχειριστής δεν έχει προβεί σε εγκατάσταση πρόσθετου λογισμικού και αλλαγής των ρυθμίσεων.

Ο χρήστης έχει εισέλθει στην εφαρμογή SkypeDust με έγκυρο όνομα χρήστη και κωδικό και αφού επιτευχθεί η είσοδος, θα μεταφερθεί στην οθόνη με τις επαφές, εκεί θα μπορέσει να προσθέσει μια επαφή (σχήμα 3.7) και αργότερα να την ενεργοποιήσει ώστε να μπορεί να χρησιμοποιήσει την εφαρμογή μέσω Skype. Αφού ο χρήστης δεχτεί την πρόσκληση θα μπορέσει να δώσει εντολές στην επαφή του SkypeDust.

Ο χρήστης με το που δεχτεί την πρόσκληση θα είναι μία από της επαφές του λογαριασμού που είναι υπεύθυνος για τον SkypeDust. Ο διαχειριστής θα μπορεί να προσθέσει μέχρι και 7450 επαφές όπου είναι και το όριο των επαφών σε ένα λογαριασμό Skype.



Σχήμα 3.7: Προσθήκη επαφής μέσω SkypeDust.



Σχήμα 3.8: Αίτημα προσθήκης στο Skype.

Για παράδειγμα αν ο χρήστης θέλει να δει τους κόμβους που έχουμε, μπορεί να δώσει στη συνομιλία του μέσου Skype την εντολή

```
list nodes
```

Για να δει τις δυνατότητες του συστήματος μπορεί να πληκτρολογήσει στη συνομιλία την εντολή

```
list capabilities
```

Αργότερα μπορεί να ελέγξει μια από τις ιδιότητες του κόμβου πληκτρολογώντας

```
node node1 light 1
```

Με αυτόν τον τρόπο μπορεί να πάρει την πρώτη εγγραφή από το Überdust. Υπάρχουν περιπτώσεις όπου ο χρήστης θέλει να ενημερώνεται όταν μια ιδιότητα αλλάζει. Για αυτό θα πληκτρολογήσει

```
register node node1 light
```

Οι παραπάνω εντολές προϋποθέτουν ότι ο χειριστής του συστήματος έχει προχωρήσει στην ονοματολογία κάποιον κόμβων αλλά και ιδιοτήτων. Βέβαια ο χρήστης μπορεί να δώσει εντολές οι οποίες περιέχουν την πραγματική ονομασία του κόμβου αλλά και τις ιδιότητες.

```
node urn:wisebed:ctitestbed:0x14d4 urn:wisebed:ctitestbed:node:capability:light
```

3.3.2 Εκτέλεση εφαρμογής με χρήση πρόσθετου λογισμικού

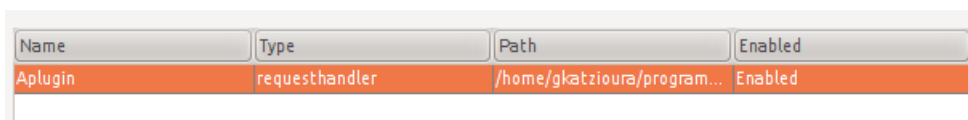
Μέσα στην εφαρμογή έχουμε εγκαταστήσει κάποια αρχεία πρόσθετου λογισμικού (plugins). Ο χρήστης κατευθύνεται στην οθόνη επιλογής πρόσθετου λογισμικού και ενεργοποιεί ένα από τα κομμάτια που είναι εγκατεστημένα. Στη περίπτωση μας το πρόσθετο λογισμικό το οποίο ενεργοποίησε, είναι ρυθμισμένο ώστε σε όποιο μήνυμα έρθει από τον χρήστη να δώσει μια αυτοματοποιημένη απάντηση.

Για παράδειγμα εάν ο χρήστης μέσω Skype δώσει την εντολή

```
list nodes
```

θα πάρει την απάντηση

```
Oct 6, 2012 4:20:49 PM: Default reply Message!
```



Name	Type	Path	Enabled
Aplugin	requesthandler	/home/gkatzioura/program...	Enabled

Σχήμα 3.9: Ενεργοποιημένο Πρόσθετο λογισμικό.

```
16:20:49 EEST] Emmanouil: list nodes
16:20:49 EEST] Gkatziouras Emmanouil: Oct 6, 2012 4:20:49 PM: Default reply Message!
```

Σχήμα 3.10: Αυτόματη απάντηση μέσω Πρόσθετου λογισμικού.

Κεφάλαιο 4

Η υλοποίηση του SkypeDust

Στο κεφάλαιο αυτό δίνεται μια πλήρης περιγραφή της υλοποίησης του SkypeDust. Περιγράφεται η αρχιτεκτονική και οι τεχνολογίες που χρησιμοποιήθηκαν για το κάθε επίπεδο. Επίσης καταγράφονται οι μέθοδοι και οι τεχνικές προγραμματισμού του κάθε επιπέδου. Τέλος αναφέρεται η ροή λειτουργίας του λογισμικού και η αλληλεπίδραση μεταξύ των επιπέδων της αρχιτεκτονικής.

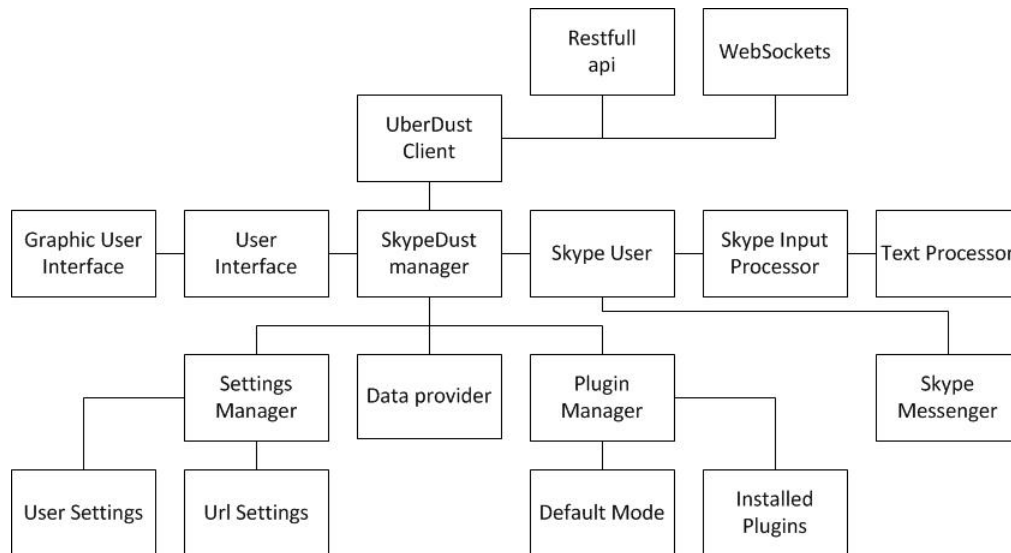
4.1 Περιγραφή αρχιτεκτονικής

Η αρχιτεκτονική του SkypeDust αποτελείται από πολλά επίπεδα τα οποία συνεργάζονται μεταξύ τους για την ομαλή λειτουργία της εφαρμογής. Για την υλοποίηση κάθε επιπέδου της αρχιτεκτονικής χρησιμοποιήθηκαν τεχνικές και τρόποι σχεδιασμού εδραιωμένοι στον χώρο του αντικειμενοστραφούς προγραμματισμού. Τα κυρία επίπεδα της αρχιτεκτονικής (σχήμα 4.1) είναι ο διαχειριστής SkypeDust, η διεπαφή χρήστη, ο διαχειριστής ρυθμίσεων, ο πάροχος δεδομένων, ο χρήστης Überdust, ο χρήστης Skype και ο διαχειριστής πρόσθετου λογισμικού. Κάθε ένα από αυτά τα επίπεδα έχει τα δικά του υποεπίπεδα.

Ο διαχειριστής SkypeDust (SkypeDust manager) παίζει σημαντικό ρόλο σε κρίσιμες αποφάσεις όσον αφορά τη λειτουργία του SkypeDust. Είναι ένα κομβικό σημείο το οποίο συνδέει τα μέρη του λογισμικού.

Η διεπαφή χρήστη (User Interface) είναι υπεύθυνη για την αλληλεπίδραση χρήστη-λογισμικού. Μέσω του γραφικού περιβάλλοντος και τους τρόπους αλληλεπίδρασης ο χρήστης μπορεί να πάρει αποφάσεις για τη λειτουργία του λογισμικού.

Ο διαχειριστής ρυθμίσεων (Settings Manager) είναι υπεύθυνος για την αποθήκευση ρυθμίσεων κρίσιμων όσο αφορά την λειτουργία του προγράμματος οι οποίες καθορίζουν επιλογές όπως για παράδειγμα το λογαριασμό Skype.



Σχήμα 4.1: Διάγραμμα αρχιτεκτονικής

Ο πάροχος δεδομένων (Data provider) είναι υπεύθυνος για δεδομένα που αφορούν την εφαρμογή όπως η ονοματολογία κόμβων και ιδιοτήτων.

Ο χρήστης Überdust (Überdust Client) είναι υπεύθυνος για την διαχείριση αιτημάτων στο Überdust και την έκδοση εντολών.

Ο χρήστης Skype (Skype User) είναι υπεύθυνος για τις λειτουργίες που αφορούν την διαχείριση υπηρεσιών του Skype δηλαδή την αλληλεπίδραση της χρήσης εφαρμογής μέσω Skype.

Ο διαχειριστής πρόσθετου λογισμικού (Plugin Manager) είναι υπεύθυνος για την διαχείριση και την παροχή του πρόσθετου λογισμικού που έχει εγκαταστήσει ο χρήστης.

4.2 Ο Διαχειριστής SkypeDust

Ο διαχειριστής SkypeDust είναι ο πυρήνας του λογισμικού. Είναι υπεύθυνος για την αρχικοποίηση του λογισμικού, για τις αποφάσεις που θα παρθούν κατά την διάρκεια εκτέλεσης όπως και της σύνδεσης μεταξύ των επιπέδων. Όταν ξεκινάει η εφαρμογή ο διαχειριστής SkypeDust αναλαμβάνει να επικοινωνήσει με το διαχειριστή ρυθμίσεων ώστε να πάρει τα στοιχεία του λογαριασμού Skype που είναι ήδη αποθηκευμένος ώστε η μόνη περίπτωση που ο χρήστης θα θελήσει να στείλει στοιχεία λογαριασμού να είναι σε περίπτωση που θέλει να αλλάξει λογαριασμό. Έπειτα αναλαμβάνει να ανοίξει την επικοινωνία με τον πάροχο δεδομένων ώστε να μπορεί να ζητήσει τα δεδομένα που χρειάζεται όταν ο χρή-

στης εισέλθει επιτυχώς στο σύστημα. Επιπλέον αφού έχει λάβει τις ρυθμίσεις μπορεί να δώσει στον πελάτη Überdust τις διευθύνσεις του διακομιστή ώστε να είναι σε ετοιμότητα η επικοινωνία με το Überdust. Η διεπαφή χρήστη αρχικά έχει πάρει από τον διαχειριστή SkypeDust το λογαριασμό που είναι ήδη αποθηκευμένος παρόλα αυτά ο χρήστης έχει την επιλογή να δώσει τα στοιχεία άλλου λογαριασμού.

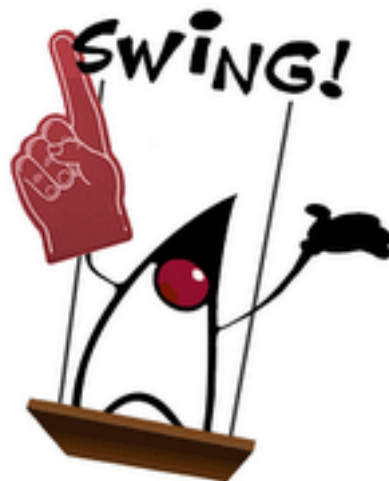
Αυτά τα στοιχεία μεταφέρονται στον διαχειριστή SkypeDust. Αυτά τα στοιχεία είναι που θα δοθούν στη διεπαφή χρήστη εάν η είσοδος ήταν πετυχημένη ή όχι. Οι ενέργειες όμως που γίνονται είναι πολύ περισσότερες. Πρώτον ο διαχειριστής στέλνει αυτά τα δεδομένα στον διαχειριστή ρυθμίσεων ώστε να αποθηκευτούν για την επόμενη φορά αλλά επίσης τα στέλνει και στον Skype χρήστη ώστε να μπορέσει να εισέλθει στο σύστημα και να αρχίσει μια συνεδρία με το Skype. Ο χρήστης Skype θα δώσει την απάντηση στο αν είναι έγκυρες οι ρυθμίσεις και αν έχει γίνει η είσοδος στο Skype. Ο διαχειριστής SkypeDust το αποτέλεσμα θα το μεταφέρει στη διεπαφή χρήστη και αναλόγως ο χρήστης θα μπορέσει να συνεχίσει στην εφαρμογή αν είναι σωστά τα στοιχεία ή όχι.

Τα δεδομένα που χρειάζεται να επιδείξει η διεπαφή χρήστη τα λαμβάνει από τον πάροχο δεδομένων μέσω του διαχειριστή. Δηλαδή ο διαχειριστής αναλαμβάνει να δώσει τα δεδομένα για την ονοματολογία των κόμβων, την ονοματολογία των ιδιοτήτων όπως και τους λογαριασμούς που επιτρέπονται να χρησιμοποιούν την εφαρμογή μέσω του Skype. Με αυτόν τον τρόπο τα όποια προβλήματα που μπορεί να προκύψουν κατά τη διάρκεια προσπέλασης δεδομένων τα αναλαμβάνει ο διαχειριστής SkypeDust και έπειτα στο επίπεδο της διεπαφής χρήστη μεταφέρεται η πληροφορία που ενδιαφέρει το χρήστη και αν αυτή υπάρχει. Επίσης ο διαχειριστής με το που πραγματοποιηθεί η είσοδος στο σύστημα αναλαμβάνει να επικοινωνήσει με τον διαχειριστή πρόσθετου κώδικα ώστε να πάρει τα πρόσθετα κομμάτια κώδικα που έχει εγκαταστήσει και έχει ήδη επιλέξει ο χρήστης να λειτουργήσουν.

Κατά την διάρκεια εκτέλεσης της εφαρμογής ο διαχειριστής SkypeDust αναλαμβάνει να προωθεί τα αιτήματα που στέλνονται για εγκατάσταση ή χρησιμοποίηση πρόσθετου κομματιού κώδικα στο διαχειριστή πρόσθετου κώδικα. Σε περίπτωση που υπάρχει αίτημα για χρήση συγκεκριμένου κομματιού κώδικα αναλαμβάνει να το φορτώσει και στα αντίστοιχα επίπεδα. Επίσης προωθεί τα αιτήματα για αποθήκευση και αλλαγή δεδομένων στον πάροχο δεδομένων, και φροντίζει να ενημερώνει τον πάροχο και τα διάφορα επίπεδα για τους χρήστες οι οποίοι πλέον μπορούν να έχουν πρόσβαση στην εφαρμογή μέσω Skype ή το αντίθετο. Τέλος αναλαμβάνει και την προώθηση αιτημάτων προσθήκης φίλου (ή διαγραφής) στο επίπεδο του εκτελέσιμου Skype.

4.3 Η Διεπαφή Χρήστη

Για την διεπαφή χρήστη έγινε χρήση της Swing εργαλειοθήκης της Java. Η εργαλειοθήκη Swing δημιουργήθηκε ώστε να παρέχει ένα μεγαλύτερο εύρος από κομμάτια γραφικής διεπαφής χρήστη από την αφηρημένη εργαλειοθήκη παραθύρων της Java (Abstract Window Toolkit-AWS). Το Swing παρέχει μια εμφάνιση πιο κοντά στα γραφικό περιβάλλον της κάθε πλατφόρμας αλλά υποστηρίζει και την δικιά του ανεξάρτητη εμφάνιση. Το τμήματα του Swing είναι υλοποιημένα εξ ολοκλήρου σε Java ώστε να εξασφαλίζεται η ανεξάρτητη λειτουργία από την πλατφόρμα πάνω στην οποία τρέχει η Java εφαρμογή.



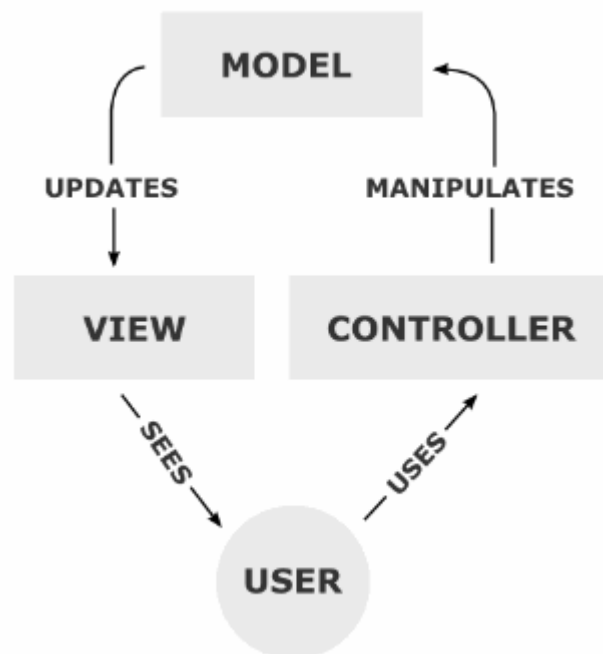
Σχήμα 4.2: Λογότυπο Java Swing

Η άλλη επιλογή που υπήρχε ήταν η χρήση της JavaFx. Η JavaFx είναι μια πλατφόρμα λογισμικού η οποία δίνει την δυνατότητα για ανάπτυξη πλούσιων εφαρμογών με γραφικό περιβάλλον και πολλά εργαλεία. Ο λόγος που δεν χρησιμοποιήθηκε ήταν επειδή κατά την διάρκεια εκπόνησης της διπλωματικής δεν ήταν διαθέσιμη σε όλες τις πλατφόρμες.

Η διεπαφή χρήστη δίνει την δυνατότητα στον χρήστη να μπορεί να δίνει τις ρυθμίσεις του στο λογισμικό για την λειτουργία του, όπως και να ενισχύει την λειτουργικότητα του με εγκατάσταση πρόσθετων κομματιών λογισμικού. Περιέχει μια οθόνη διαχείρισης λογαριασμού για επαφές που θέλει να προσθέσει και επαφές που θέλει να καταργήσει από τον λογαριασμό που χρησιμοποιεί όπως και ποιες από αυτές τις επαφές θα μπορούν να χρησιμοποιούν την επαφή. Περιέχει μια οθόνη ώστε να μπορεί να εκδώσει ονοματολογίες για κόμβους και ιδιότητες, να τις αλλάξει όπως και να τις καταργήσει. Τέλος παρέχει μια οθόνη

στην οποία ο χρήστης μπορεί να διαχειριστεί τα πρόσθετα κομμάτια κώδικα δηλαδή να τα εγκαταστήσει, να τα καταργήσει όπως και να επιλέξει πια θα χρησιμοποιήσει ανάλογα με τις ανάγκες τις οποίες θέλει να καλύψει.

Όλες αυτές τις ενέργειες που πραγματοποιεί ο χρήστης στη διεπαφή χρήστη, η διεπαφή τις στέλνει στον διαχειριστή SkypeDust ώστε αυτός να διαχειριστεί αυτά τα αιτήματα και να δώσει αυτές τις προτιμήσεις στα επίπεδα τα οποία αντιστοιχούν. Δηλαδή όσο αφορά τις ονοματολογίες κόμβων, ιδιοτήτων όπως και τις προτιμήσεις χρήστη γίνονται οι αντίστοιχες ενέργειες στον πάροχο δεδομένων. Όσο αφορά τα επιπρόσθετα κομμάτια λογισμικού φροντίζει να γίνουν οι κατάλληλες ενέργειες στον διαχειριστή πρόσθετου λογισμικού. Τέλος η διεπαφή παρέχει πληροφορίες για το λογισμικό, τον τρόπο χρήσης του όπως και τα στοιχεία του δημιουργού του λογισμικού ώστε ο χρήστης να μπορεί να επικοινωνήσει μαζί του σε περίπτωση προβλήματος.



Σχήμα 4.3: Αρχιτεκτονική μοντέλου-ελεγκτή-όψης

Η διεπαφή χρήστη ακολουθεί την αρχιτεκτονική μοντέλου-ελεγκτή-όψης (Model-view-controller [23]). Η αρχιτεκτονική model-view-controller (σχήμα 4.3) διαχωρίζει την πληροφορία από την αλληλεπίδραση του χρήστη με αυτήν. Το μοντέλο αποτελείται από δεδομένα κανόνες και συναρτήσεις. Η όψη είναι το αποτέλεσμα της αναπαράστασης δεδομένων στο γραφικό περιβάλλον. Ο ελεγκτής αποτελείται από τις συναρτήσεις που μετατρέπουν τις δράσεις του χρήστη σε εντολές

για τα μοντέλα ή τις όψεις.

Στη περίπτωση του SkypeDust τα μοντέλα είναι απλά αντικείμενα Java (Pojo [25]) τα οποία έχουν ενθυλακωμένη την πληροφορία που θα πρέπει να αναπαρασταθεί. Απλά αντικείμενα Java χρησιμοποιούνται για την περίπτωση της ονοματολογίας των δυνατοτήτων, της ονοματολογίας των κόμβων, των ρυθμίσεων του πρόσθετου λογισμικού και των ρυθμίσεων του χρήστη.

Η όψη είναι η διεπαφή χρήστη η οποία παίρνει την πληροφορία που επιστρέφει ο ελεγκτής μέσω των μοντέλων, και την αναπαριστά στον χρήστη με την βοήθεια των εργαλείων και των συναρτήσεων που προσφέρει το Swing.

Ο ελεγκτής είναι ο διαχειριστής SkypeDust (το επίπεδο που περιγράφεται παραπάνω) ο οποίος είναι υπεύθυνος ανάλογα με τις δράσεις του χρήστη και τα αποτελέσματα που προκύπτουν από αυτές, να ενθυλακώσει την πληροφορία σε μοντέλα και να τα επιστρέψει στο επίπεδο της όψης.

Η αρχιτεκτονική μοντέλου-ελεγκτή-όψης είναι ιδιαίτερα δημοφιλής στις διαδικτυακές εφαρμογές, αλλά πλέον την υιοθετούν πολλές εφαρμογές προσωπικού υπολογιστή όπως και διάφορες εργαλειοθήκες προγραμματισμού.

Μέχρι στιγμής η διεπαφή περιλαμβάνει γραφικό περιβάλλον σε Java Swing αλλά υπάρχουν οι κατάλληλες προϋποθέσεις ώστε να μπορεί να υπάρξει και μια διεπαφή μέσω κονσόλας.

4.4 Ο Διαχειριστής Ρυθμίσεων

Ο ρόλος του διαχειριστή ρυθμίσεων είναι να αποθηκεύει και να ανακτά τα στοιχεία του λογαριασμού χρήστη αλλά και τις ρυθμίσεις όσο αφορά τη διεύθυνση του Überdust. Παρέχει μια αφηρημένη διαδικασία αποθήκευσης και ανάκτησης δεδομένων. Οι ρυθμίσεις του χρήστη όσο αφορά το όνομα χρήστη και το όνομα που φαίνεται στην επαφή Skype αποθηκεύονται σε αρχείο xml.

Η επεκτάσιμη γλώσσα σήμανσης (XML-Extensible Markup Language) προσδιορίζει κάποιους κανόνες για την κωδικοποίηση των δεδομένων ώστε να είναι δυνατή η ανάγνωση του και από τον άνθρωπο όσο και από ένα υπολογιστικό σύστημα. Οι σχεδιαστικοί στόχοι της XML είναι η έμφαση στην απλότητα τη γενικότητα και τη χρήση στο διαδίκτυο. Χρησιμοποιείται ευρέως για την αναπαράσταση δομών δεδομένων. Μια από τις κύριες χρήσεις της είναι η αποθήκευση ρυθμίσεων σε εφαρμογές.

Αρχείο Xml 4.1: Αρχείο ρυθμίσεων SkypeDust

```
<?xml version="1.0" encoding="UTF-8"?><Settings>
  <username>gkatzbot</username>
  <nickname>Emmanouil</nickname>
</Settings>
```

Ο κωδικός του χρήστη όμως αποθηκεύεται κρυπτογραφημένος χρησιμοποιώντας το προηγμένο πρότυπο κρυπτογράφησης (AES) [1]. Ο AES είναι ένας αλγόριθμος κρυπτογράφησης που προστατεύει ευαίσθητα και αταξινόμητα δεδομένα. Ο AES βασίζεται στη σχεδιαστική αρχή του δικτύου υποκατάστασης μετάθεσης (Substitution-permutation network) και είναι γρήγορος όσο αφορά το υλικό αλλά και το λογισμικό. Το κλειδί κρυπτογράφησης προσδιορίζει έναν αριθμό επαναλήψεων κρυπτογράφησης που μετατρέπουν την είσοδο στο τελικό αποτέλεσμα δηλαδή την έξοδο. Στην περίπτωση μας χρησιμοποιούμε ένα κλειδί των 128 bits, οπότε θα έχουμε 10 κύκλους επαναλήψεων.

Αρχείο Java 4.2: Κρυπτογράφηση Αποκρυπτογράφηση στο SkypeDust

```
private static final String algo = "AES";
private static final byte[] enckey = new byte[]{
    'C', 'h', 'a', 'n', 'g', 'e', ' ',
    't', 'h', 'i', 's', ' ', 'o', 'n', 'e', ' ', '!', '!' };

private void storeEncrypted(String password) {
    try {
        Key key = new SecretKeySpec(enckey, algo);
        Cipher c = Cipher.getInstance(algo);
        c.init(Cipher.ENCRYPT_MODE, key);
        byte[] encval = c.doFinal(password.getBytes());
        FileOutputStream fout = new FileOutputStream(FileManage.Passfile);
        PrintStream pstream = new PrintStream(fout);
        pstream.println(new BASE64Encoder().encode(encval));
    } catch (IOException | InvalidKeyException |
        IllegalBlockSizeException | BadPaddingException |
        NoSuchAlgorithmException | NoSuchPaddingException ex) {
        Logger.getLogger(UserAccount.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private String getEncrypted() {
    String password = null;
    try {
        Key key = new SecretKeySpec(enckey, algo);
        Cipher c = Cipher.getInstance(algo);
        DataInputStream datains = new DataInputStream(
            new FileInputStream(FileManage.Passfile));
        BufferedReader breader = new BufferedReader(
            new InputStreamReader(datains));
        String encpass = breader.readLine();
        c.init(Cipher.DECRYPT_MODE, key);
        byte[] decordval = new BASE64Decoder().decodeBuffer(encpass);
        byte[] decVal = c.doFinal(decordval);
        password = new String(decVal);
    } catch (IllegalBlockSizeException | BadPaddingException | InvalidKeyException |
        IOException | NoSuchAlgorithmException | NoSuchPaddingException ex) {
        Logger.getLogger(UserAccount.class.getName()).log(Level.SEVERE, null, ex);
    }
    return password;
}
```

Ο διαχειριστής ρυθμίσεων έχει υβριδική λειτουργία. Λειτουργεί ως ένα απλό Java αντικείμενο αλλά παράλληλα και ως ένα αντικείμενο με λειτουργίες οι οποίες

όμως δεν είναι δημόσιες σε άλλες κλάσεις. Όταν σε ένα αντικείμενο διαχειριστή ρυθμίσεων εκτελεστούν οι συναρτήσεις αποθήκευσης αλλά και ανάκτησης αντικειμένων (συνηθισμένες λειτουργίες σε ένα απλό Java αντικείμενο), τότε ο διαχειριστής ρυθμίσεων καλεί τις ιδιωτικές του συναρτήσεις οι οποίες αναλαμβάνουν την αποθήκευση, την ανάκτηση από τα αρχεία xml ή την κρυπτογράφηση και αποκρυπτογράφηση των κωδικών.

Οι ρυθμίσεις του χρήστη έχουν την αντίστοιχη λειτουργικότητα ώστε ο διαχειριστής SkypeDust απλά να στέλνει τα στοιχεία και αυτά να αποθηκεύονται όπως και όποτε τα ζητά και να γίνονται οι αντίστοιχες διαδικασίες με αποτέλεσμα ο διαχειριστής της εφαρμογής να μην έρχεται αντιμέτωπος με τις λεπτομέρειες όπως και με τα σφάλματα κατά την αποθήκευση. Με τον ίδιο τρόπο λειτουργούν και οι ρυθμίσεις διευθύνσεων μόνο που σε αυτήν την περίπτωση δεν υπάρχει η ανάγκη για κρυπτογράφηση πληροφορίας.

4.5 Ο Χρήστης Skype

Ο χρήστης Skype είναι υπεύθυνος για την διαχείριση των υπηρεσιών του Skype. Κατά την αρχικοποίηση του, δίνοντας τα στοιχεία του χρήστη αρχίζει μια συνεδρία με το Skype. Κατά την διάρκεια αυτής της συνεδρίας είναι υπεύθυνος για την λήψη μηνυμάτων από χρήστες της εφαρμογής, για την αποστολή μηνυμάτων σε αυτούς όπως και για την προσθήκη άλλων χρηστών. Όταν ο χρήστης δώσει εντολή για κλείσιμο της εφαρμογής είναι υπεύθυνος για την έξοδο του χρήστη από τον λογαριασμό και τον τερματισμό της συνεδρίας.

Ο χρήστης Skype κατά την διάρκεια της συνεδρίας στα μηνύματα που λαμβάνει εκτελεί κάποιες ενέργειες. Υπάρχουν οι προεπιλεγμένες ενέργειες που μπορεί να εκτελέσει αλλά υπάρχει και η επιλογή ο χρήστης της εφαρμογής να εγκαταστήσει ένα πρόσθετο κομμάτι λογισμικού, βάση του οποίου εκτελούνται ενέργειες διαφορετικές από τις προεπιλεγμένες. Με αυτόν το τρόπο ο χρήστης μπορεί να έχει την επιθυμητή αντίδραση σε λήψη μηνύματος από μια επαφή. Οι ενέργειες δεν είναι τίποτα άλλο από μια διαδικασία κλήσης συναρτήσεων ώστε να εκπληρωθεί το αίτημα του χρήστη. Το μήνυμα που λαμβάνεται επεξεργάζεται και δημιουργούνται τα αντίστοιχα αιτήματα που πρέπει να σταλούν στον πελάτη Überdust. Ανάλογα με την διαδικασία ο χρήστης Skype θα απαντήσει με ένα μήνυμα απευθείας που τον προειδοποιεί σε περίπτωση που το αίτημα που στέλνει είναι λάθος.

Ο χρήστης Skype επίσης αναλαμβάνει την παροχή υπηρεσίας μηνυμάτων στις επαφές του χρήστη. Δηλαδή κατά την διάρκεια εκτέλεσης της εφαρμογής κάποιο από τα επίπεδα του SkypeDust μπορεί να χρειαστεί την υπηρεσία αποστολής δεδομένων μέσω Skype για να ενημερώσει τις επαφές που χρησιμοποιούν την

επαφή αναλόγως. Ο χρήστης Skype δίνει στα άλλα επίπεδα την δυνατότητα αποστολής αυτών των μηνυμάτων.

4.6 Ο Πελάτης Überdust

Ο πελάτης Überdust αναλαμβάνει την επικοινωνία με τον αντίστοιχο Überdust διακομιστή. Οι λειτουργίες του χωρίζονται σε δύο σημεία. Στην άμεση έκδοση αιτήσεων μέσω ενός πελάτη Πρωτοκόλλου Μεταφοράς Υπερκειμένου (Http) και στην ασύγχρονη ενημέρωση αλλαγών μέσω ενός πελάτη Websocket.

Και οι δύο πελάτες πρέπει να πληρούν κάποιες προϋποθέσεις. Αυτό εξασφαλίζεται με το να είναι υποκλάσεις ενός αφηρημένου μοντέλου Überdust πελάτη ο οποίος καθορίζει τις προϋποθέσεις αυτές ώστε να είναι δυνατή η δημιουργία του πελάτη μεταφοράς υπερκειμένου αλλά και του πελάτη Websocket. Πρέπει να υπάρχει η υποδοχή για την υπηρεσία αποστολής μηνυμάτων μέσω Skype, η υποδοχή για τον χρόνο καθυστέρησης σε απάντηση όπως και η δυνατότητα αποκωδικοποίησης εντολών και κόμβων.

Στη περίπτωση της άμεσης έκδοσης αιτήσεων ο Überdust πελάτης χρησιμοποιεί έναν αντίστοιχο πελάτη Http ο οποίος εκδίδει αιτήσεις στο Restfull api του Überdust. Αυτές οι αιτήσεις μπορεί να είναι είτε για ενημέρωση της κατάστασης του περιβάλλοντος σε ένα δωμάτιο του κτηρίου είτε για την αλλαγή σε μία από τις καταστάσεις του. Η απάντηση έρχεται σε αναπαράσταση javascript αντικειμένου (JSON [19]) το οποίο ο πελάτης καλείται να επεξεργαστεί και να στείλει μέσω του Skype χρήστη μια φιλική απάντηση, στην επαφή που εξέδωσε το αίτημα χρησιμοποιώντας το χρήστη SkypeDust.

Αρχείο Bash 4.3: Αίτηση Rest στο Restfull api του Überdust για τις τελευταίες 5 μετρήσεις led του κόμβου urn:wisebed:ctitestbed:0x122 σε μορφή Json

```
server="überdust.cti.gr";
anode="urn:wisebed:ctitestbed:0x122";
acapability="urn:wisebed:node:capability:led";

wget $server/rest/testbed/1/node/$anode/capability/$acapability/json/limit/5

cat 5
{"nodeId":"urn:wisebed:ctitestbed:0x122",
 "readings":[
  {"timestamp":1350051823000,"reading":0,"stringReading":""},
  {"timestamp":1350051813000,"reading":0,"stringReading":""},
  {"timestamp":1350051713000,"reading":0,"stringReading":""},
  {"timestamp":1350051678000,"reading":0,"stringReading":""},
  {"timestamp":1350051668000,"reading":0,"stringReading":""}],
 "capabilityId":"urn:wisebed:node:capability:led"
}
```

Υπάρχει η επιλογή της άμεσης απάντησης και ενημέρωσης που είναι προκαθορισμένη. Όμως ο χρήστης έχει την επιλογή στο να χρησιμοποιήσει ένα πρόσθετο κομμάτι λογισμικού ώστε η ενέργεια που θα προβεί ο πελάτης να είναι η επιθυμητή.

Για τις ανάγκες της υλοποίησης του Rest πελάτη χρησιμοποιήθηκε η βιβλιοθήκη Apache commons HttpClient [13]. Η Java παρέχει μια πρότυπη βιβλιοθήκη για πρόσβαση σε πληροφορίες μέσω του Http πρωτοκόλλου (Hyper-text Transfer Protocol), όμως δεν παρέχει την λειτουργικότητα και την ευελιξία που χρειάζεται η εφαρμογή SkypeDust. Η βιβλιοθήκη HttpClient παρέχει έναν αποτελεσματικό και σύγχρονο τρόπο για την δημιουργία ενός Http πελάτη με βάση τα σύγχρονα πρότυπα και συστάσεις, για τις εφαρμογές που απαιτούν την υλοποίηση ενός πελάτη Http.

Αρχείο Java 4.4: Αίτημα Http get μέσω HttpClient στην εφαρμογή SkypeDust

```
private String basicGet(String getUrl){  
    try {  
        HttpClient httpClient = new DefaultHttpClient();  
        HttpGet httpGet = new HttpGet(getUrl);  
        HttpResponse httpResponse = httpClient.execute(httpGet);  
        HttpEntity httpEntity = httpResponse.getEntity();  
        if (httpEntity != null)  
            return EntityUtils.toString(httpEntity);  
    } catch (IOException ex) {  
        Logger.getLogger(RestfullClient.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    return "404";  
}
```

Στη περίπτωση ασύγχρονης ενημέρωσης αλλαγών ο Überdust χρησιμοποιεί έναν Websocket πελάτη ο οποίος εγγράφεται για άμεση ενημέρωση σε έναν Websocket διακομιστή που διαθέτει ο Überdust. Ο πελάτης μπορεί να εγγραφεί στην αλλαγή μιας από τις ιδιότητες του κόμβου. Σε περίπτωση που συμβεί μια αλλαγή της ιδιότητας σε έναν κόμβο ο Überdust διακομιστής στέλνει τις ενημερώσεις σε όλους πελάτες έχουν εγγραφεί για ενημέρωση. Ο πελάτης Überdust λαμβάνει την ενημέρωση και είναι υπεύθυνος να στείλει το περιεχόμενο της ενημέρωσης δηλαδή την τιμή της ιδιότητας πίσω στην επαφή με τον προκαθορισμένο τρόπο μέσω του SkypeDust πελάτη. Και σε αυτήν την περίπτωση αντί να ακολουθήσει την προκαθορισμένη διαδικασία ο χρήστης μπορεί να εγκαταστήσει πρόσθετο λογισμικό το οποίο να ακολουθεί διαφορετικές ενέργειες από την προκαθορισμένη.

Αρχείο Java 4.5: Παράδειγμα εγγραφής σε websocket για συνεχή ενημέρωση για τις μεταβολές σε μια ιδιότητα κόμβου

```
package eu.uber dust . applications . listenerws ;  
  
import eu.uber dust . communication . protobuf . Message ;
```



```
import eu.uber dust . communication . websocket . readings . WSReadingsClient ;
import org . apache . log4j . Logger ;
import org . apache . log4j . PropertyConfigurator ;
import java . util . Observable ;
import java . util . Observer ;

public class ListenerApp implements Observer {

    private static final Logger LOGGER = Logger . getLogger ( ListenerApp . class ) ;

    private static ListenerApp ourInstance = null ;
    private String server ;
    private String protocol ;

    public static ListenerApp getInstance () {
        synchronized ( ListenerApp . class ) {
            if ( ourInstance == null ) {
                ourInstance = new ListenerApp () ;
            }
        }
        return ourInstance ;
    }

    public ListenerApp () {
        PropertyConfigurator . configure (
            this . getClass () . getClassLoader () . getResource ( " log4j . properties " ) ) ;
        server = " ws : // localhost : 8081 / uber dust / readings . ws " ;

        WSReadingsClient . getInstance () . setServerUrl ( server ) ;
        WSReadingsClient . getInstance () . subscribe (
            " urn : wisebed : ctitestbed : 0x7061 " ,
            " urn : wisebed : node : capability : pir " ) ;

        WSReadingsClient . getInstance () . addObserver ( this ) ;
    }

    @Override
    public void update ( Observable o , Object arg ) {
        if ( !( o instanceof WSReadingsClient ) ) {
            return ;
        }
        if ( arg instanceof Message . NodeReadings ) {
            Message . NodeReadings reading = ( Message . NodeReadings ) arg ;
            LOGGER . info ( reading . getReading ( 0 ) . getNode () ) ;
        }
    }
}
```

4.7 Ο Διαχειριστής Πρόσθετου Λογισμικού

Ο διαχειριστής πρόσθετου λογισμικού είναι το επίπεδο που παρέχει υπηρεσίες όσο αφορά το πρόσθετο λογισμικό που εγκαθιστά ο χρήστης.

Η μια λειτουργία του είναι η εγκατάσταση του πρόσθετου λογισμικού. Όταν μέσω της διεπαφής χρήστη προστίθεται λογισμικό, ο διαχειριστής SkypeDust στέλνει το αντίστοιχο λογισμικό στο διαχειριστή πρόσθετου λογισμικού ώστε

να ελέγξει αν πληρεί τις προϋποθέσεις για να εγκατασταθεί και αργότερα να χρησιμοποιηθεί.

Η δεύτερη λειτουργία είναι η επιστροφή του λογισμικού που χρειάζονται τα άλλα επίπεδα στις διάφορες περιπτώσεις. Ο διαχειριστής ελέγχει το λογισμικό το οποίο είναι ήδη εγκατεστημένο και ρυθμισμένο να χρησιμοποιηθεί έναντι του προκαθορισμένου λογισμικού. Επίσης ελέγχει το είδος λογισμικού που ζητείται. Σε περίπτωση που δεν έχει εγκατασταθεί και δεν έχει επιλεγεί πρόσθετο λογισμικό τότε ο διαχειριστής επιστρέφει το προκαθορισμένο λογισμικό.

Όσον αφορά τη χρήση ενός πλαισίου λογισμικού η γλώσσα Java παρέχει πολλές λύσεις μέσω βιβλιοθηκών ανοιχτού λογισμικού. Χαρακτηριστικά παραδείγματα οι βιβλιοθήκες Osgi και Jpf. Λόγω των απλών λειτουργιών που χρειάζεται το λογισμικό και των μικρών απαιτήσεων όσο αφορά τα πρόσθετα κομμάτια λογισμικού προχωρήσαμε στη δημιουργία ενός δικού μας πλαισίου πρόσθετου λογισμικού. Η ανάπτυξή του βασίστηκε στη δυνατότητα της Java για δυναμική φόρτωση μεταγλωττισμένων κλάσεων κατά την διάρκεια λειτουργίας του προγράμματος.

Αρχείο Java 4.6: Δυναμική φόρτωση κλάσης με χρήση Java Reflection

```
URLClassLoader classLoader = URLClassLoader.newInstance(  
    new URL[] { "path/javaClasses.jar" } );  
Class<?> aclass = classLoader.load( "com/example/MyClass" );  
AclassInterface bclass = (AclassInterface) aclass.newInstance();
```

Η φόρτωση κλάσης απαιτεί την ύπαρξη μιας κλάσης η οποία βασίζεται σε μια διεπαφή Java του προγράμματος. Για να είναι επιτυχής η φόρτωση κλάσης πρέπει η κλάση να ακολουθεί τις προδιαγραφές που ορίζει η επιλεγμένη διεπαφή. Αργότερα το πρόγραμμα θα έχει τη δυνατότητα να δημιουργήσει ένα στιγμιότυπο της κλάσης που έχει φορτωθεί. Αυτή η κλάση θα έχει τις επιθυμητές λειτουργίες όσο αφορά τις μεθόδους που θα καλεί το πρόγραμμα από αυτήν.

4.8 Ο Πάροχος Δεδομένων

4.8.1 Τρόποι Αποθήκευσης Δεδομένων

Ο πάροχος δεδομένων είναι υπεύθυνος για την αποθήκευση και παροχή δεδομένων στα διάφορα επίπεδα. Κατά την διάρκεια εκτέλεσης της εφαρμογής δημιουργούνται δεδομένα τα οποία διαχειρίζεται η εφαρμογή. Λόγω της αποθήκευσης δεδομένων, αυτά τα δεδομένα ανακτώνται από την εφαρμογή και μετά τον τερματισμό της.

Η αποθήκευση της πληροφορίας επιτυγχάνεται με διάφορους τρόπους. Ένας τρόπος είναι η αποθήκευση δεδομένων στο σκληρό δίσκο σε μορφή απλού κειμένου. Σε αυτή τη περίπτωση ο προγραμματιστής καλείται να αναπτύξει από

μόνος του ένα μηχανισμό για την αποθήκευση και την ανάκτηση. Όσο όμως αυξάνονται οι απαιτήσεις της εφαρμογής και ο όγκος δεδομένων, απαιτείται ένας πιο αποτελεσματικός τρόπος για την αποθήκευση και ανάκτηση πληροφορίας ο οποίος να εγγυάται την ταχύτητα αλλά και την ευελιξία στη χρήση του.

Μια από της επιλογές είναι η αποθήκευση δεδομένων σε μορφή xml καθώς υπάρχουν οι αντίστοιχες βιβλιοθήκες λογισμικού οι οποίες εγγυώνται την αποτελεσματική αποθήκευση και ανάκτηση δεδομένων. Μία άλλη λύση είναι η λύση που χρησιμοποιήθηκε για τις ανάγκες της εφαρμογής, και είναι η χρήση μιας σχεσιακής βάσης δεδομένων [2]. Με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων ηλεκτρονικά αποθηκευμένης πληροφορίας. Πέρα από την ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων παρέχει τα αποκαλούμενα συστήματα διαχείρισης βάσης δεδομένων και τη δυνατότητα γρήγορης άντλησης και ανανέωσης δεδομένων. Ένα σύστημα διαχείρισης βάσης δεδομένων είναι το λογισμικό μέσω του οποίου γίνεται η δημιουργία, η διαχείριση, η συντήρηση και η χρήση μιας ηλεκτρονικής βάσης δεδομένων. Το λογισμικό χρησιμοποιεί μεθόδους καταλογοποίησης, ανάκτησης, και εκτέλεσης ερωτημάτων σχετικών με τα δεδομένα.

4.8.2 Η βάση δεδομένων του SkypeDust

Ο πάροχος δεδομένων χρησιμοποιεί την ενσωματωμένη βάση δεδομένων που παρέχει η γλώσσα προγραμματισμού Java, τη βάση δεδομένων Apache Derby [3]. Με το όρο ενσωματωμένη βάση δεδομένων εννοούμε ένα σύστημα διαχείρισης βάσης δεδομένων το οποίο είναι αυστηρά ενσωματωμένο στο λογισμικό της εφαρμογής το οποίο ζητά πρόσβαση στα δεδομένα. Απαιτεί λίγη ή μη συνεχόμενη διατήρηση και ο τελικός χρήστης της εφαρμογής έχει πρόσβαση στα δεδομένα μόνο μέσω της εφαρμογής. Η Apache Derby είναι μια μηχανή δομημένης γλώσσας ερωτημάτων (SQL [30]) πλήρως γραμμένη σε Java και διατίθεται δωρεάν κάτω από την άδεια χρήσης Apache. Στο λογισμικό μας η μηχανή derby παρέχει λειτουργίες δημιουργίας εγγραφής, ανανέωσης και διαγραφής πινάκων (CRUD operations [8]).

Κατά την αρχικοποίηση του συστήματος φροντίζει για την δημιουργία της βάσης και των αντίστοιχων πινάκων εάν αυτό δεν έχει γίνει. Οι πίνακες αφορούν δεδομένα όπως η ονοματολογία των κόμβων και ιδιοτήτων, τις επαφές που μπορούν να χρησιμοποιούν το λογισμικό μέσω Skype όπως και τα πρόσθετα κομμάτια λογισμικού που είναι εγκατεστημένα και τι σκοπό εξυπηρετούν. Μία άλλη λειτουργία είναι να αποθηκεύει η να διαγράφει τις επαφές Skype οι οποίες είναι εγγεγραμμένες για αλλαγές τιμών σε ιδιότητες κόμβων. Ο πάροχος δεδομένων

χρησιμοποιείται από το διαχειριστή SkypeDust για την αποστολή αντίστοιχων πληροφοριών τόσο στη διεπαφή χρήστη όσο και στον χρήστη Skype.

Ο πάροχος δεδομένων είναι ένα αντικείμενο πρόσβασης δεδομένων (Dao [10]). Με τη χρήση ενός αντικειμένου πρόσβασης δεδομένων, οι ενέργειες ανάκτησης και αποθήκευσης δεδομένων γίνονται χωρίς την επαφή σε λεπτομέρειες που αφορούν τη βάση δεδομένων. Τα πλεονεκτήματα αυτής της τεχνικής είναι ότι μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές ανεξαρτήτου μεγέθους, κρύβει όλες τις τεχνικές λεπτομέρειες που αφορούν τη βάση δεδομένων, οι αλλαγές στο μηχανισμό της βάσης δεδομένων είναι πιο εύκολες και είναι πιο εύκολο για όλα τα επίπεδα του λογισμικού να έχουν πρόσβαση στα δεδομένα.

Ο πάροχος δεδομένων υλοποιήθηκε έτσι ώστε οι μέθοδοι πάνω στη βάση δεδομένων όπως η εκτέλεση ερωτημάτων ή η διαγραφή πινάκων να είναι ορατές μόνο στο επίπεδο του παρόχου δεδομένων. Τα αποτελέσματα αυτών των ενεργειών ο πάροχος δεδομένων τα επεξεργάζεται και επιστρέφει την πληροφορία σε μορφή Java αντικειμένων ή επιστρέφει μηνύματα που δηλώνουν την επιτυχία ή την αποτυχία της ενέργειας που ζητήθηκε από ένα άλλο επίπεδο.

Ο πελάτης Überdust λόγω του ότι είναι βασισμένος στο μηχανισμό πρόσθετου λογισμικού το οποίο μπορεί να αλλάξει ανά πάσα στιγμή ανάλογα με τις προτιμήσεις του χρήστη, και λόγω των απαιτήσεων του μπορεί να καλέσει τον πάροχο δεδομένων απευθείας ανά πάσα στιγμή.

Αρχείο Java 4.7: Java κλάση με λειτουργίες δημιουργίας εγγραφής διαγραφής σε Apache Derby

```
public class EmbeddedDatabase {

    private String driver = "org.apache.derby.jdbc.EmbeddedDriver";
    private String protocol = "jdbc:derby";
    private String dbName = "embeddedDB";
    private Connection connection;
    private Statement statement;
    private static final String table_person = "PERSON";
    private static final String createtablePerson =
        "CREATE TABLE PERSON(NAME VARCHAR(100),NICKNAME VARCHAR(100))";
    private static final String droptablePerson = "DROP TABLE PERSON";

    public EmbeddedDatabase(){

        try {
            Class.forName(driver);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
        }

        try {
            connection = DriverManager.getConnection(protocol+":"+dbName+";create=true");
            createTables();
        } catch (SQLException ex) {
            Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
        }

        try {
            statement = connection.createStatement();
            statement.executeUpdate(createtablePerson);
        } catch (SQLException ex) {
            Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
        }

        try {
            statement.executeUpdate(droptablePerson);
        } catch (SQLException ex) {
            Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

}
```

```

private void createTables() {
    try {
        statement = connection.createStatement();
        statement.execute(createTablePerson);
        statement.close();
    } catch (SQLException ex) {
        Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void dropTables() {
    try {
        statement = connection.createStatement();
        statement.execute("DROP TABLE PERSON");
        statement.close();
    } catch (SQLException ex) {
        Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public int insertPerson(String name, String nickname) {
    int toret = 0;
    try {
        statement = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        ResultSet resultSet = statement.executeQuery(
            "SELECT*FROM PERSON WHERE NAME='"+name+"' OR NICKNAME='"+nickname+"'");
        toret = statement.executeUpdate(
            "INSERT INTO PERSON VALUES('"+name+"','"+nickname+"')");
        statement.close();
    } catch (SQLException ex) {
        Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
    }
    return toret;
}

public String[][] getPersons() {
    try {
        statement = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        ResultSet resultSet = statement.executeQuery("SELECT*FROM PERSON");
        List<String[]> toret = new ArrayList<String[]>();
        while(resultSet.next())
            toret.add(new String[]{resultSet.getString("NAME"),resultSet.getString(
                "NICKNAME")});
        return toret.toArray(new String[toret.size()][]);
    } catch (SQLException ex) {
        Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);
    }
    return null;
}

public void deletePerson(String name) {
    try {
        statement = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
        ResultSet resultSet = statement.executeQuery(
            "SELECT*FROM PERSON "+ "WHERE NAME='"+name+"'");
        while(resultSet.next())
            resultSet.deleteRow();
        resultSet.close();
    }
}

```

```
        statement.close();  
    } catch (SQLException ex) {  
        Logger.getLogger(EmbeddedDatabase.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

4.9 Ροή Λειτουργίας

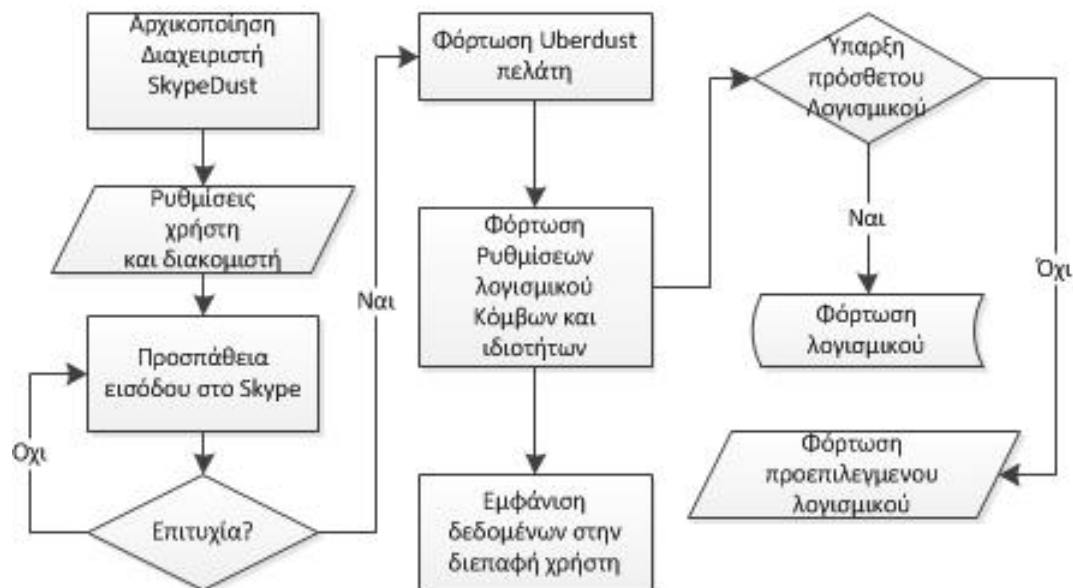
Ο χρήστης ανοίγει την εφαρμογή. Με το που ανοίγει η εφαρμογή, ξεκινά ο διαχειριστής SkypeDust την αρχικοποίηση του προγράμματος. Αρχικά μέσω του χρήστη Skype δημιουργεί μια συνεδρία με το Skype και ανοίγει την επικοινωνία με τον πάροχο δεδομένων. Έπειτα επικοινωνεί με τον διαχειριστή ρυθμίσεων, εάν υπάρχουν αποθηκευμένες ρυθμίσεις επιστρέφονται στον διαχειριστή SkypeDust ο οποίος και μεταφέρει τις πληροφορίες στην αρχική οθόνη ώστε ο χρήστης να μη χρειαστεί να εισάγει από την αρχή το λογαριασμό και τον κωδικό. Ο χρήστης έχει την επιλογή να εισάγει στην είσοδο άλλο λογαριασμό. Αφού ο χρήστης πατήσει για να εισέλθει στο σύστημα τότε ο διαχειριστής SkypeDust στέλνει τα δεδομένα στον χρήστη Skype ώστε να πραγματοποιηθεί η είσοδος στο Skype. Σε περίπτωση λανθασμένων πιστοποιητικών η συνεδρία παραμένει ανοιχτή και περιμένει για το επόμενο όνομα χρήστη και κωδικό ώστε να πραγματοποιηθεί η είσοδος.

Μετά την επιτυχημένη είσοδο στο σύστημα ο διαχειριστής SkypeDust αφού πάρει την διεύθυνση για το Überdust από το διαχειριστή ρυθμίσεων, δημιουργεί έναν πελάτη Überdust. Ελέγχει μέσω του πάροχου δεδομένων άμα υπάρχει πρόσθετο λογισμικό εγκατεστημένο και επιλεγμένο για λειτουργία αλλιώς χρησιμοποιεί το προεπιλεγμένο λογισμικό. Ο διαχειριστής έπειτα φορτώνει στον χρήστη Skype τον πελάτη Überdust όπως και το πρόσθετο λογισμικό το οποίο θα χρησιμοποιηθεί.

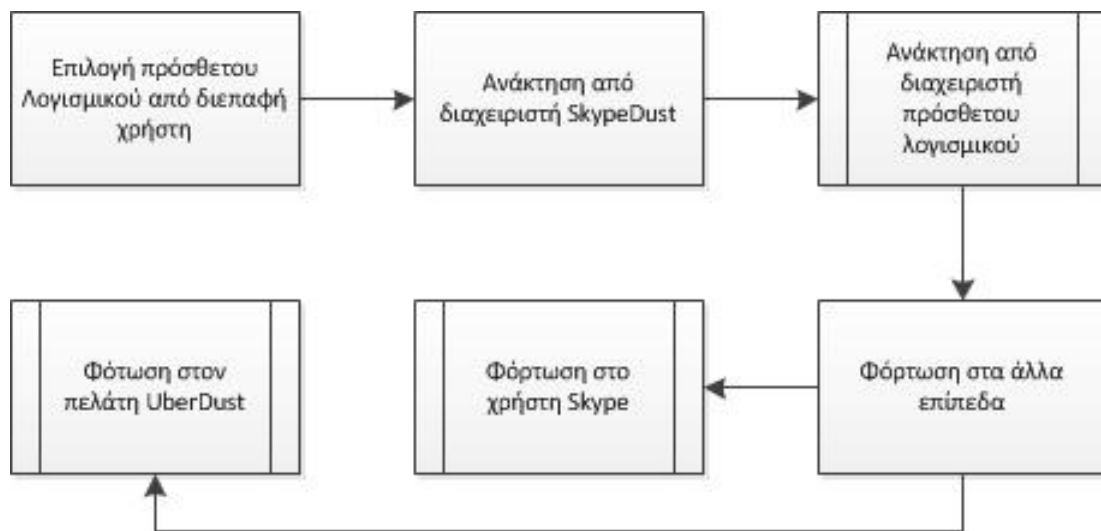
Μετά το πέρας αυτών των διαδικασιών ο διαχειριστής SkypeDust αναλαμβάνει να στείλει δεδομένα που αφορούν την εφαρμογή στη διεπαφή χρήσης. Ανακτά τις επαφές του λογαριασμού μέσω του πελάτη Skype και ποιες από αυτές τις επαφές επιτρέπονται να χρησιμοποιούν την εφαρμογή και τα στέλνει στην οθόνη του λογαριασμού του χρήστη. Μετά μέσω του πάροχου δεδομένων ανακτά τις ονοματολογίες των κόμβων και ιδιοτήτων και τις στέλνει στην σελίδα των ονοματολογιών κόμβου. Τέλος παίρνει τις πληροφορίες για το πρόσθετο λογισμικό που είναι εγκατεστημένες, τις ρυθμίσεις του και στέλνει τα δεδομένα στη σελίδα του πρόσθετου λογισμικού.

Αν κατά την διάρκεια λειτουργίας του λογισμικού ο χρήστης αποφασίσει να κάνει μια αλλαγή είτε στις επαφές είτε στην ονοματολογία των κόμβων και ιδιο-

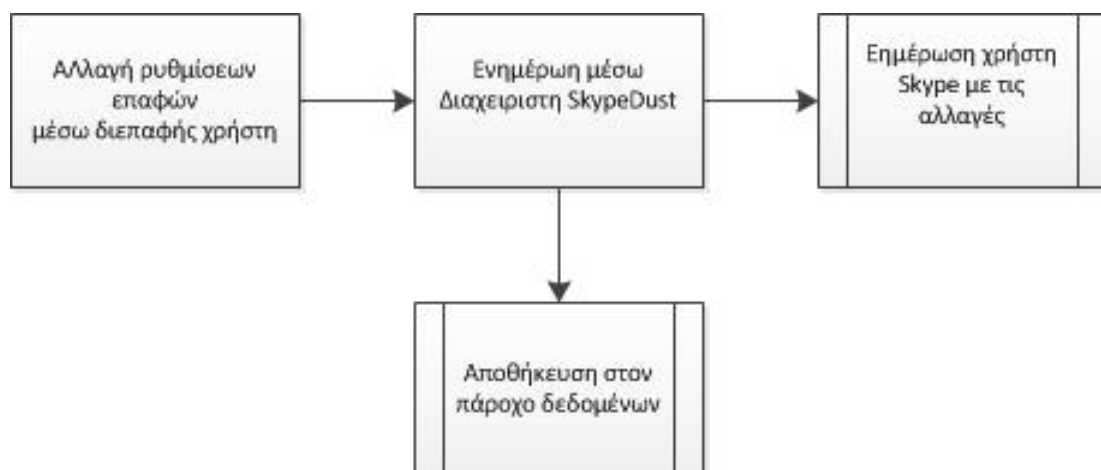
τήτων όπως και στο πρόσθετο λογισμικό, ο διαχειριστής λογισμικού αναλαμβάνει να φορτώσει αυτές τις αλλαγές απευθείας στα άλλα επίπεδα χωρίς να χρειαστεί ο χρήστης να επανεκκινήσει την εφαρμογή.



Σχήμα 4.4: Ροή προγράμματος.



Σχήμα 4.5: Επιλογή πρόσθετου λογισμικού κατά την διάρκεια εκτέλεσης.



Σχήμα 4.6: Αλλαγή ρυθμίσεων επαφών κατά την διάρκεια εκτέλεσης.

Κεφάλαιο 5

Χρήση πρόσθετου λογισμικού

Στο κεφάλαιο αυτό παρουσιάζεται η χρήση πρόσθετου λογισμικού στην εφαρμογή. Γίνεται αναφορά στις προϋποθέσεις που πρέπει να πληρεί το πρόσθετο λογισμικό ώστε να είναι δυνατόν να εγκατασταθεί και να χρησιμοποιηθεί από την εφαρμογή. Έπειτα αναφέρονται οι κατηγορίες πρόσθετου λογισμικού τις οποίες μπορεί να χρησιμοποιήσει ο χρήστης και πια θα είναι η επίδραση τους στη ροή λειτουργίας του προγράμματος. Τέλος δίνονται κάποια παραδείγματα χρήσης πρόσθετου λογισμικού.

5.1 Προϋποθέσεις

5.1.1 Γενικές Προϋποθέσεις

Τα αρχεία πρόσθετου λογισμικού που μπορεί να εγκαταστήσει ένας χρήστης στο σύστημα πρέπει να έχουν συγκεκριμένη μορφή και οπωσδήποτε να πληρούν κάποιες προϋποθέσεις. Η πρώτη βασική προϋπόθεση είναι να είναι ένα αρχείο σε μορφή συμπιεσμένου αρχείου Java (JAR). Μέσα σε αυτό το αρχείο Java θα πρέπει να περιέχεται ένα αρχείο XML με τις ρυθμίσεις του και ένα αρχείο .class το οποίο ουσιαστικά είναι το λογισμικό (bytecode) που θέλουμε να εγκαταστήσουμε στο SkypeDust. Το αρχείο .class είναι ένα αρχείο κλάσης Java το οποίο έχει μεταγλωττιστεί(compiled). Λόγω του ότι το λογισμικό SkypeDust είναι γραμμένο σε Java το πρόσθετο λογισμικό θα πρέπει να είναι και αυτό γραμμένο σε Java. Το αρχείο Java δεν είναι τίποτα άλλο παρά το αποτέλεσμα της μεταγλώττισης και συμπίεσης του αρχείου Java μαζί με το αρχείο Settings.xml. Σε πρώτη φάση πρέπει οπωσδήποτε να υπάρχουν αυτά τα αρχεία αλλιώς υπάρχει σφάλμα κατά την εγκατάσταση του λογισμικού.

Το Settings.xml αρχείο θα πρέπει να περιέχει πληροφορίες για το πρόσθετο κομμάτι κώδικα όπως και για το δημιουργό του. Το πρώτο στοιχείο αφορά την

ονομασία του λογισμικού και πως θα εμφανίζεται στον χρήστη, το δεύτερο στοιχείο είναι το όνομα του προγραμματιστή, το τρίτο ο τρόπος επικοινωνίας μαζί του όπως και η ιστοσελίδα του. Τα επόμενα τρία στοιχεία αφορούν γενικές πληροφορίες όπως την άδεια λογισμικού, την έκδοση και την περιγραφή. Τα δύο τελευταία στοιχεία είναι και τα πιο βασικά. Ο διαχειριστής πρόσθετου λογισμικού από εκεί θα αποφασίσει τι είδους λογισμικό είναι και σε ποια περίπτωση του SkypeDust αντιστοιχεί, όπως και πως θα φορτώσει το λογισμικό. Το στοιχείο `type` δηλώνει τι τύπος είναι το πρόσθετο λογισμικό και σε ποια λειτουργία του SkypeDust απευθύνεται. Το επόμενο στοιχείο διαφέρει ανά τύπο και δηλώνει που βρίσκεται μέσα στο αρχείο Java το αρχείο `.class`. Ο διαχειριστής πρόσθετου λογισμικού είναι υπεύθυνος για να βρεί σε ποια κατηγορία ανήκει το πρόσθετο λογισμικό. Για αυτό από το xml αρχείο με τις ρυθμίσεις θα ελέγξει το στοιχείο `type`.

Αρχείο 5.1: Παράδειγμα Settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Settings>
  <name>Plugin Name</name>
  <author>Gkatziouras Emmanouil</author>
  <email>gkatzioura@gmail.com</email>
  <url>http://egkatzioura.wordpress.com</url>
  <license>Gpl V 2.0</license>
  <version>1.0</version>
  <description>A Plugin example</description>
  <type>requesthandler</type>
  <requesthandlerclass>aplugin/Aplugin</requesthandlerclass>
</Settings>
```

5.2 Οι κατηγορίες

Τα πρόσθετα κομμάτια λογισμικού χωρίζονται σε τρεις κατηγορίες. Η πρώτη είναι το λογισμικό που διαχειρίζεται τα μηνύματα που έρχονται από τις επαφές του Skype και αναλαμβάνει να πάρει απόφαση για το ποιες ενέργειες θα διεξαχθούν. Έτσι δίνεται η δυνατότητα για διαμόρφωση των εντολών που μπορεί να στέλνει ο χρήστης μέσω του Skype. Επίσης δίνεται η δυνατότητα η αντίδραση στο μήνυμα που θα ληφθεί να διαφέρει ανά επαφή και συγκεκριμένες επαφές να έχουν πρόσβαση σε κάποιες ιδιότητες και σε άλλες όχι. Αμέσως δημιουργούνται πολλοί συνδυασμοί όσο αφορά τα αιτήματα των επαφών και τις αυτόματες απαντήσεις που παίρνουν. Παραδείγματος χάρη ένας προγραμματιστής θα μπορούσε να δημιουργήσει έναν ολόκληρο μηχανισμό ο οποίος να βασίζεται στο ιστορικό των αιτημάτων, ποιες επαφές ήταν ενεργές και να ακολουθεί διαφορετική διαδικασία ανάλογα με την περίπτωση.

Η δεύτερη και τρίτη κατηγορία αφορά την αντίδραση του Überdust πελάτη

αφού έχει πάρει και επεξεργαστεί τα δεδομένα. Η πρώτη κατηγορία αφορά τον Restfull πελάτη όπου έχουμε σύγχρονο τρόπο λειτουργίας. Από την στιγμή που ο πελάτης παίρνει τα αποτελέσματα του αιτήματος του από το Überdust τότε στη προεπιλεγμένη περίπτωση μπορεί απλά να στείλει τα δεδομένα πίσω στον χρήστη. Στη περίπτωση μας όμως μπορεί να επεξεργαστεί αυτά τα δεδομένα και ανάλογα με τις προτιμήσεις του να αναπτύξει ειδικά σενάρια λειτουργίας. Δηλαδή αν μία ιδιότητα είναι μικρότερη από μια συγκεκριμένη τιμή ανά δύο ώρες να ενημερώνεται για την κατάσταση και να δώσει το SkypeDust εντολή για αλλαγή στο Überdust την επόμενη μέρα.

Το ίδιο ακριβώς γίνεται και για την περίπτωση που ο πελάτης χρησιμοποιεί τα websockets. Η διαφορά είναι ότι στα websockets ο πελάτης απλά ενημερώνεται ανά τακτά διαστήματα και δεν μπορεί να αλληλεπιδράσει με το Überdust ώστε να αλλάξει η τιμή μιας ιδιότητας.

Από τα παραπάνω προκύπτει ένας τεράστιος αριθμός συνδυασμών με αποτέλεσμα ο χρήστης να μπορεί να έχει ακριβώς την λειτουργικότητα που θέλει όσο αφορά το SkypeDust. Το λογισμικό για αυτοματοποιημένο έλεγχο κτηρίου μπορεί να είναι εγκατεστημένο σε ένα μεγάλο αριθμό κτηρίων με τις δικές του ιδιαιτερότητες το καθένα. Αυτό οδηγεί στο λογισμικό να διαφέρει ως προς τις υπηρεσίες που παρέχει. Το SkypeDust μπορεί να επικοινωνήσει και να αντεπεξέλθει σε αυτές τις διαφορές λόγω του μηχανισμού πρόσθετου κώδικα και στο ότι το SkypeDust αποτελεί ένα δομοστοιχειωμένο λογισμικό.

5.2.1 Διαχείριση Μηνυμάτων

Αρχείο Settings 5.2: Περίπτωση Διαχείρισης Μηνυμάτων

```
<type>requesthandler</type>  
<requesthandlerclass>aplugin/Aplugin</requesthandlerclass>
```

Η κλάση πρέπει να υλοποιεί συγκεκριμένες μεθόδους οι οποίες την βοηθάνε στη λειτουργικότητα της. Ποιο συγκεκριμένα πρέπει να έχει υποδοχή για τις επιτρεπόμενες επαφές καθώς είναι και αυτές από τις οποίες θα λαμβάνει μηνύματα, υποδοχή για τον Übedust πελάτη με τον οποίο θα μπορεί στέλνει αιτήσεις στο Überdust και έναν αποστολέα μηνυμάτων Skype για να μπορεί να ενημερώσει τις επαφές για τα αιτήματά τους και πληροφορίες για την ροή λειτουργίας.

Αρχείο Class 5.3: Συναρτήσεις Διαχείρισης Μηνυμάτων

```
public void setAllowedContacts(String [] CommandCons)  
public void setUberClient(RestfullClient uberClient);  
public void setWebsocketClient(SkypedustWebsocket skypedustWebsocket);  
public void run();
```

```
public String inputParse(String author, String body);  
public void setSkypeMessenger(SkypeMessenger skypeMessenger);
```

5.2.2 Διαχείριση Restfull Πελάτη

Αρχείο Settings 5.4: Περίπτωση Διαχείρισης Restfull αποτελεσμάτων

```
<type>resthandler</type>  
<resthandlerclass>aplugin/Aplugin</resthandlerclass>
```

Για τον Restfull πελάτη χρειαζόμαστε μια υποδοχή για τα αποτελέσματα από την απάντηση του Überdust , μια υποδοχή για τον πελάτη Überdust σε περίπτωση αποστολής άλλων αιτημάτων και υποδοχή για τον αποστολέα μηνυμάτων Skype. Τέλος υπάρχει μια συνάρτηση η οποία καλείται για να τρέξει το λογισμικό αφού έχει αρχικοποιηθεί.

Αρχείο Class 5.5: Συναρτήσεις Διαχείρισης Restfull αποτελεσμάτων

```
public void setCapabilityResults(String [][] results);  
public void run();  
public void setSkypeMessenger(SkypeMessenger skypeMessenger);
```

5.2.3 Διαχείριση Ασύγχρονων Ενημερώσεων

Αρχείο Settings 5.6: Περίπτωση Διαχείρισης Ασύγχρονων Ενημερώσεων

```
<type>notificationhandler</type>  
<notificationhandlerclass>aplugin/Aplugin</notificationhandlerclass>
```

Οι υποδοχές για την διαχείριση ασύγχρονων ενημερώσεων είναι παρόμοιες με την περίπτωση του Restfull πελάτη. Εδώ έχουμε μια υποδοχή για τα αποτελέσματα από την απάντηση του Überdust με αντίστοιχη μορφή, μια υποδοχή για τον πελάτη Überdust σε περίπτωση αποστολής άλλων αιτημάτων ,μια υποδοχή για τον αποστολέα μηνυμάτων Skype όπως και τη συνάρτηση η οποία καλείται για να τρέξει το λογισμικό αφού έχει αρχικοποιηθεί.

Αρχείο Class 5.7: Συναρτήσεις Διαχείρισης Ασύγχρονων Ενημερώσεων

```
public void setNodeReadings(Message.NodeReadings nodeReadings);  
public void setSkypeMessenger(SkypeMessenger skypeMessenger);  
public void run();
```

5.2.4 Παραδείγματα Πρόσθετου Λογισμικού

Έστω ο χρήστης θέλει συγκεκριμένες ώρες το SkypeDust στα αιτήματα άλλων χρηστών να δίνει μια αυτόματη απάντηση τότε μπορεί να φορτώσει πρόσθετο λογισμικό. Έστω αυτό το μήνυμα πρέπει να μεταδίδεται μεταξύ 15:00 και 18:00.

Αρχείο Class 5.8: Αυτόματη απάντηση

```
@Override
public String inputParse(String string, String string1) {

    try {
        SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
        Long first = dateFormat.parse("15:00").getTime();
        Long last = dateFormat.parse("18:00").getTime();
        Long now = new Date().getTime();

        if (first < now & now < last) {

            return "Default reply Message!";
        }
    } catch (ParseException ex) {
        Logger.getLogger(Aplugin.class.getName()).log(Level.SEVERE, null, ex);
    }
    return "problem";
}
```

Ακόμα υπάρχει περίπτωση όπου κάποια επαφή που δεν έχει πρόσβαση στο σύστημα να προσπαθήσει να δώσει εντολή. Τότε το SkypeDust πρέπει να του πετάξει μήνυμα λάθους άλλα και να προειδοποιήσει τις άλλες επαφές για αυτό το συμβάν.

Αρχείο Class 5.9: Μη Αποδεκτή Επαφή

```
@Override
public String inputParse(String author, String message) {

    if (!Arrays.asList(allowedContacts).contains(author)) {

        String warning_message = "Unauthorized user tried to issue command";
        skypeMessenger.sendMessage(allowedContacts, warning_message);

        return "You are unauthorized";
    }
    else {
        return "Your request is being processed";
    }
}
```


Κεφάλαιο 6

Συμπεράσματα και προοπτικές

6.1 Συμπεράσματα

Φτάνοντας στο τέλος της διπλωματικής εργασίας μπορούμε να καταλήξουμε σε κάποια συμπεράσματα όσον αφορά την εμπειρία ανάπτυξης λογισμικού που αλληλεπιδρά με ένα ολοκληρωμένο σύστημα ελέγχου κτηρίου.

Η ανάπτυξη ενός τέτοιου είδους λογισμικού αποτελεί από μόνο του μεγάλη πρόκληση. Ένας από τους λόγους είναι ότι αποτελεί μέρος ενός ευρύτερου συστήματος με πολλές ιδιαιτερότητες και απαιτήσεις. Το λογισμικό πρέπει να είναι προσαρμοσμένο σε αυτές τις απαιτήσεις ώστε να εξασφαλίζεται η ομαλή του λειτουργία και ως επακόλουθο να είναι χρήσιμο αλλά και μια ευχάριστη εμπειρία για το χρήστη. Είναι ευτύχημα ότι η τεχνολογία που έχει ήδη αναπτυχθεί από την μεριά του διακομιστή δεδομένων *Überdust* και της διαδικτυακής υπηρεσίας που προσφέρει διευκολύνει κατά πολύ τη διαδικασία ανάπτυξης. Η δυνατότητα της διεπαφής προγραμματισμού εφαρμογών που προσφέρει ο *Überdust* ήταν κρίσιμη στο να μπορέσει ο προγραμματιστής να έχει την ελευθερία επιλογών κατά την διάρκεια ανάπτυξης λογισμικού. Οι περιορισμοί από μέρους του *Überdust* ήταν λίγοι με αποτέλεσμα ο φοιτητής να μπορεί να βάλει την δική του λογική πάνω στην υλοποίηση αναπτύσσοντας το δικό του λογισμικό.

Η χρήση των υπηρεσιών *Skype* αποτελούσε και αυτή από μόνη της μια πρόκληση. Ένας λόγος είναι ότι ο προγραμματιστής καλείται να δημιουργήσει λογισμικό το οποίο πρέπει να πληρεί κάποιες τεχνικές προϋποθέσεις ώστε να είναι εφικτή η χρήση αυτών των υπηρεσιών. Επίσης το *Skype* ως λογισμικό έχει κάποιους περιορισμούς όσο αφορά την χρήση των υπηρεσιών του. Δεν δίνεται η δυνατότητα της πλήρους διαχείρισης του λογαριασμού του χρήστη όπως και της πλήρους διαχείρισης των υπηρεσιών που παρέχει. Η υπηρεσία φωνητικών και βίντεο κλήσεων διαχειρίζεται από το εκτελέσιμο αρχείο *Skype runtime* χω-

ρίς ο προγραμματιστής να μπορεί να επεξεργαστεί τα φωνητικά δεδομένα και να προχωρήσει περαιτέρω στην ανάπτυξη υπηρεσιών.

Μια άλλη πρόκληση αποτελούσε η επιλογή των κατάλληλων εργαλείων για την υλοποίηση του λογισμικού. Είναι ξεκάθαρο ότι με τα κατάλληλα εργαλεία η ανάπτυξη του λογισμικού μπορεί να γίνει ταχύτερη χωρίς το μέγεθος των απαιτήσεων να παίζει μεγάλο ρόλο. Το γραφικό περιβάλλον Swing έχει πολλά εργαλεία ώστε να διευκολύνεται η ανάπτυξη μια εφαρμογής από τον χρήστη με αποτέλεσμα μια εφαρμογή πλούσια σε λειτουργικότητα χωρίς ο προγραμματιστής να καταφύγει σε επίπονες διαδικασίες. Ο προγραμματιστής μπορεί με ένα εργαλείο σχεδιασμού να κάνει μια εφαρμογή με γραφικό περιβάλλον πάνω στο οποίο μπορούν να προστεθούν στοιχεία (widgets), τα οποία μπορούν να προσφέρουν τις λειτουργίες που χρειάζονται για την ομαλή λειτουργία της εφαρμογής.

Η βάση δεδομένων Apache Derby λύνει το πρόβλημα της αποθήκευσης και ανάκτησης δεδομένων. Αυτό έχει ως αντίκτυπο να μην χρειαστεί να δημιουργηθεί η ανάπτυξη ενός μηχανισμού διαχείρισης δεδομένων, ο οποίος πιθανότατα δεν θα είχε την ίδια ευελιξία και λειτουργικότητα με την Apache Derby.

Σημαντικό ρόλο στην ανάπτυξη καλού κώδικα έπαιξε η χρήση των ενσωματωμένων περιβαλλόντων προγραμματισμού (IDE [15]). Τα IDE δίνουν την δυνατότητα της εύκολης και ταχείας ανάπτυξης λογισμικού όπως και της σωστής διαχείρισης των βιβλιοθηκών που μπορεί να χρησιμοποιεί ένας χρήστης. Επιπλέον τα IDE παρέχουν ειδικές λειτουργίες όσο αφορά την χρήση συγκεκριμένων εργαλείων πχ ένας visual designer για εφαρμογές με γραφικό περιβάλλον.

Τέλος μια μεγάλη πρόκληση ήταν η αρχιτεκτονική του λογισμικού. Η επιλογή της κατάλληλης αρχιτεκτονικής παίζει τεράστιο ρόλο για την εύκολη ανάπτυξη του λογισμικού ειδικά σε περιπτώσεις που προστίθενται ή αλλάζουν απαιτήσεις. Μια σωστή αρχιτεκτονική δίνει την δυνατότητα αλλαγής κώδικα σε συγκεκριμένα κομμάτια του προγράμματος χωρίς να χρειαστεί να αλλάξουν άλλα κομμάτια του προγράμματος. Λόγω του ότι έχουμε μια εφαρμογή που επικοινωνεί με έναν διακομιστή αλλά και χρησιμοποιεί λογισμικό όπως το SkypeKit arι, είναι κρίσιμο να μπορούν να γίνονται εύκολα αλλαγές στον πηγαίο κώδικα. Όταν υπάρχει το ενδεχόμενο αλλαγών σε μία από τις διεπαφές που μπορεί να προσφέρει το UberDust αλλά και το SkypeKit αμέσως οδηγούν και σε αλλαγή του πηγαίου κώδικα. Επίσης χάρη στην αρχιτεκτονική είναι δυνατή η παρουσία μηχανισμού διαχείρισης πρόσθετου λογισμικού που να μπορεί εύκολα να αλληλεπιδρά με τα άλλα πεδία της αρχιτεκτονικής. Τέλος είναι πιο εύκολη η τροποποίηση του λογισμικού από τρίτους και με προοπτικές για χρησιμοποίηση για άλλα παρόμοια συστήματα αυτομάτου ελέγχου.

Σημαντική παράμετρος στην ανάπτυξη του λογισμικού ήταν το γεγονός ότι το σύστημα Uberdust είναι ένα σύστημα το οποίο χρησιμοποιείται από το Ερευνητικό Ακαδημαϊκό Ινστιτούτο Τεχνολογίας Υπολογιστών. Η ανάπτυξη αλλά και

οι δοκιμές της εφαρμογής έγιναν με βάση ένα σύστημα το οποίο ήδη λειτουργούσε σε καθημερινή βάση κάτι το οποίο βοήθησε στην ανάπτυξη ενός λογισμικού που τρέχει σε αληθινό σενάριο.

6.2 Μελλοντικοί Στόχοι

Ένας από τους στόχους είναι στο μέλλον η εφαρμογή να μπορέσει να εκμεταλλευτεί πλήρως τις υπηρεσίες που παρέχει το Skype. Θα είναι κάτι πολύ σημαντικό όσο αφορά την ευκολία στη χρήση της εφαρμογής αλλά και για την αύξηση των τρόπων χρήσης. Αυτό εξαρτάται κατά ένα μεγάλο ποσοστό από τις δυνατότητες που το Skypekit api θα δώσει στους προγραμματιστές στις επόμενες εκδόσεις. Μέχρι στιγμής οι δυνατότητες συνομιλίας και κλήσεις video διαχειρίζονται από κλειστό λογισμικό και δεν δίνεται η δυνατότητα επεξεργασίας του πχ. voice-to-text.

Άμεσο στόχο αποτελεί η δημιουργία μιας πλούσιας βιβλιοθήκης σε μεθόδους και δυνατότητες, ώστε να διευκολυνθεί η διαδικασία ανάπτυξης πρόσθετου λογισμικού. Το αποτέλεσμα θα είναι η ανάπτυξη πρόσθετου λογισμικού για το SkypeDust να διευκολύνεται από ήδη έτοιμες μεθόδους χωρίς ο προγραμματιστής να μπαίνει στη διαδικασία ανάπτυξης καινούργιων μεθόδων.

Βιβλιογραφία

- [1] **Advanced Encryption Standard.** http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- [2] **Database** <http://en.wikipedia.org/wiki/Database>
- [3] **Apache Derby.** http://en.wikipedia.org/wiki/Apache_Derby
- [4] **Application Programming Interface** http://en.wikipedia.org/wiki/Application_programming_interface
- [5] **Building Automation** http://en.wikipedia.org/wiki/Building_automation
- [6] **Creating a GUI With JFC/Swing.** <http://docs.oracle.com/javase/tutorial/uiswing/index.html>
- [7] **Class (computer programming)** [http://en.wikipedia.org/wiki/Class_\(computer_programming\)](http://en.wikipedia.org/wiki/Class_(computer_programming))
- [8] **Create, read, update and delete** http://en.wikipedia.org/wiki/Create,_read,_update_and_delete
- [9] **Design Patterns: Elements of Reusable Object-Oriented Software.**
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
http://en.wikipedia.org/wiki/Design_Patterns
- [10] **Data access object** http://en.wikipedia.org/wiki/Data_access_object
- [11] **Effective Java: Programming Language Guide.**
Joshua Bloch
http://books.google.gr/books/about/Effective_Java.html?id=ZZOiqZQIbRMC&redir_esc=y
- [12] **Graphical User Interface.** http://en.wikipedia.org/wiki/Graphical_user_interface

- [13] **HttpClient.** <http://hc.apache.org/httpclient-3.x/>
- [14] **Hypertext Transfer Protocol.** http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [15] **Integrated development environment.** http://en.wikipedia.org/wiki/Integrated_development_environment
- [16] **Inter Process Communication** http://en.wikipedia.org/wiki/inter-process_communication
- [17] **Jar** [http://en.wikipedia.org/wiki/JAR_\(file_format\)](http://en.wikipedia.org/wiki/JAR_(file_format))
- [18] **Java** [http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [19] **Json** <http://en.wikipedia.org/wiki/JSON>
- [20] **Machine to Machine** http://en.wikipedia.org/wiki/Machine_to_machine
- [21] **Making P-Space Smart: Integrating IoT technologies in a Multi-office environment.** link.springer.com/chapter/10.1007/978-3-642-366660-4_3
- [22] **Microcontroller** <http://en.wikipedia.org/wiki/Microcontroller>
- [23] **Model-View-Controller** <http://en.wikipedia.org/wiki/Model-view-controller>
- [24] **Software Plug-ins.** http://en.wikipedia.org/wiki/Plug-in_%28computing%29
- [25] **Plain old Java object** http://en.wikipedia.org/wiki/Plain_Old_Java_Object
- [26] **Representational state transfer.** http://en.wikipedia.org/wiki/Representational_state_transfer
- [27] **Skype.** <http://en.wikipedia.org/wiki/Skype>
- [28] **SkypeDust.** <https://github.com/Uberdust/SkypeDust/wiki/SkypeDust>
- [29] **SkypeKit.** <https://developer.skype.com/public/skypekit/>
- [30] **Sql Language.** <http://en.wikipedia.org/wiki/SQL>
- [31] **Strategy Pattern** http://en.wikipedia.org/wiki/Strategy_pattern
- [32] **Swing toolkit** [http://en.wikipedia.org/wiki/Swing_\(Java\)](http://en.wikipedia.org/wiki/Swing_(Java))
- [33] **Transmission Control Protocol** http://en.wikipedia.org/wiki/Transmission_Control_Protocol

-
- [34] Überdust. <https://github.com/Überdust/webapp/wiki>
- [35] Voice over Ip http://en.wikipedia.org/wiki/Voice_over_IP
- [36] WebSockets. <http://en.wikipedia.org/wiki/WebSocket>
- [37] Wireless Sensor Network http://en.wikipedia.org/wiki/Wireless_sensor_network
- [38] XML. <http://en.wikipedia.org/wiki/XML>