



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

---

**Ολοκληρωμένο Σύστημα Ασφάλειας Βασισμένο σε  
Δίκτυα Αισθητήρων :  
Σχεδιασμός – Ανάπτυξη**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

ΤΟΥ

**ΔΗΜΗΤΡΙΟΥ ΑΝΑΓΝΩΣΤΟΠΟΥΛΟΥ**

Καθηγητής: Γεώργιος Σταμούλης  
Επιβλέπων: Δρ. Παναγιώτης Κίικιρας

Βόλος, Οκτώβριος 2008



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ  
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 6696/1  
Ημερ. Εισ.: 29-10-2008  
Δωρεά: Συγγραφέα  
Ταξιθετικός Κωδικός: ΠΤ – ΜΗΥΤΔ  
2008  
ΑΝΑ



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

---

**Ολοκληρωμένο Σύστημα Ασφάλειας Βασισμένο σε  
Δίκτυα Αισθητήρων :  
Σχεδιασμός – Ανάπτυξη**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΔΗΜΗΤΡΙΟΥ ΑΝΑΓΝΩΣΤΟΠΟΥΛΟΥ

Καθηγητής: Γεώργιος Σταμούλης  
Επιβλέπων: Δρ. Παναγιώτης Κίικιρας

Βόλος, Οκτώβριος 2008

## ΠΡΟΛΟΓΟΣ

---

Η παρούσα διπλωματική εργασία εκπονήθηκε κατά τους μήνες Φεβρουάριο – Οκτώβριο 2008 . Επιπρόσθετα , οι δοκιμές και τα αποτελέσματα έγιναν στο εργαστήριο Αισθητήρων του ΤΜΗΥΤΔ .

Αντικείμενο της εργασίας είναι η σχεδίαση και ανάπτυξη εφαρμογής Java σε γραφικό περιβάλλον που αλληλεπιδρά με δίκτυα αισθητήρων με παράλληλη αποστολή δεδομένων σε μια κεντρική βάση προς επεξεργασία και λήψη αποφάσεων, με σκοπό την διατήρηση εσωτερικών χώρων σε ασφαλή κατάσταση .

Υπεύθυνος κατά την εκπόνηση της διπλωματικής ήταν ο Καθηγητής κ. Γ. Σταμούλης στον οποίο οφείλω ιδιαίτερες ευχαριστίες για την ανάθεση αυτής και την δυνατότητα που μου δόθηκε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής Δρ. Π. Κίικρα για την υποστήριξη και την καθοδήγηση που μου παρείχε κατά την εκπόνηση της εργασίας.

Δημήτριος Αναγνωστόπουλος

## ΠΕΡΙΛΗΨΗ

---

Τα δίκτυα αισθητήρων χρησιμοποιούνται σήμερα σε ευρεία κλίμακα για την παρακολούθηση χώρων, τον έλεγχο της κυκλοφοριακής κίνησης (αέρος - εδάφους), τη συλλογή δεδομένων και την αντιμετώπιση εκτάκτων αναγκών, τη παρακολούθηση κλιματικών μεταβολών και αντιμετώπιση των φυσικών κινδύνων , στο Σύστημα Υγείας με καινοτόμες διαγνωστικές, απεικονιστικές και θεραπευτικές προσεγγίσεις κ.α .

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη και η υλοποίηση ενός ολοκληρωμένου συστήματος ασφαλείας το οποίο αλληλεπιδρά με δίκτυα αισθητήρων και ο χρήστης έχει την δυνατότητα να επεξεργάζεται τα δεδομένα που συλλέγονται από τους αισθητήρες και αποστέλλονται στη βάση, ώστε να αντιμετωπίζει εγκαίρως τις έκτακτες ανάγκες που στην συγκεκριμένη περίπτωση μας είναι η πιθανότητα πυρκαγιάς , πλημμύρας καθώς επίσης και η μη εξουσιοδοτημένη πρόσβαση στο χώρο στον οποίο αναφερόμαστε.

Πρόκειται λοιπόν για μια εφαρμογή για εσωτερικούς χώρους , κυρίως σπίτια , αποθήκες υλικού , σχολεία, επιχειρήσεις κλπ , που αποστολή έχει την συνεχή επιτήρηση του χώρου και διατήρηση του σε ασφαλή κατάσταση.

Αρχικά, γίνεται αναφορά στα απαραίτητα εργαλεία πάνω στα οποία στηρίχτηκε και αναπτύχθηκε αυτή η εφαρμογή , εν συνεχεία στα στάδια ανάπτυξης της και τέλος στη παρουσίαση του γραφικού περιβάλλοντος και των λειτουργιών που παρέχει .

## ABSTRACT

---

Nowadays, Sensor Networks are applied for a wide range of usages such as, for the surveillance of areas, for controlling aerial and road traffic, for the collection of data in order to confront emergencies, for the observation of the climatic change and to face natural disasters, and regarding the Health Service these devices can contribute to innovative diagnostic methods, as well as curing processes.

The target is to achieve an Integrated Security System (I.S.S) that will interact with sensor networks so that the user will have the ability to elaborate the data, which is collected by these sensors and sent out to the general base, in order to confront emergencies on time concerning fire, floods, and an unauthorized access to the area that we are referring.

The I.S.S. is applied in internal spaces mainly referring to houses, store rooms, schools, enterprises e.t.c and aims at a constant surveillance of the area, as well as maintaining the security of the area.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

---

<b>1. Εισαγωγή</b>	07
1.1. Αντικείμενο και στόχος διπλωματικής εργασίας	08
1.2. Δομή τεύχους διπλωματικής εργασίας	09
1.3. Φάσεις εκπόνησης διπλωματικής εργασίας	10
<b>2. Εργαλεία Ανάπτυξης Εφαρμογής</b>	12
2.1. Εισαγωγή	13
2.2. NetBeans IDE 6.1	13
2.3. MySQL server 5.0	13
2.4. Java Media Framework (JMF)	13
<b>3. Στάδια Ανάπτυξης Εφαρμογής</b>	14
3.1. Εξαγωγή και Ανάλυση Απαιτήσεων	15
3.2. Σχεδίαση – Κωδικοποίηση της Εφαρμογής	16
3.2.1 Σχεδίαση	16
3.2.2 Κωδικοποίηση	18
3.3. Δοκιμή της Εφαρμογής	26
3.4. Παρουσίαση της Εφαρμογής	29
3.4.1 Ενέργειες πριν την Εκκίνηση	29
3.4.2 Τρόποι εκκίνησης	29
3.4.3 Παρουσίαση Γραφικού Περιβάλλοντος	30
<b>4. Συμπεράσματα- Προοπτικές</b>	44
4.1. Συμπεράσματα - Προοπτικές	45
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	46

---

# ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>

## ΕΙΣΑΓΩΓΗ

---



## 1.1. Αντικείμενο και στόχος της διπλωματικής εργασίας

Τα δίκτυα αισθητήρων χρησιμοποιούνται σήμερα σε ευρεία κλίμακα για την παρακολούθηση χώρων, τον έλεγχο της κυκλοφοριακής κίνησης (αέρος - εδάφους), τη συλλογή δεδομένων και την αντιμετώπιση εκτάκτων αναγκών, τη παρακολούθηση κλιματικών μεταβολών και αντιμετώπιση των φυσικών κινδύνων , στο Σύστημα Υγείας με καινοτόμες διαγνωστικές, απεικονιστικές και θεραπευτικές προσεγγίσεις κ.α .

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη και η υλοποίηση ενός ολοκληρωμένου συστήματος ασφαλείας το οποίο αλληλεπιδρά με δίκτυα αισθητήρων και ο χρήστης έχει την δυνατότητα να επεξεργάζεται τα δεδομένα που συλλέγονται από τους αισθητήρες και να αντιμετωπίζει εγκαίρως τις έκτακτες ανάγκες που στην συγκεκριμένη περίπτωση μας είναι η πιθανότητα πυρκαγιάς , πλημμύρας καθώς επίσης και η μη εξουσιοδοτημένη πρόσβαση στο χώρο στον οποίο αναφερόμαστε.

Πρόκειται λοιπόν για μια εφαρμογή για εσωτερικούς χώρους , κυρίως σπίτια , αποθήκες υλικού , σχολεία, επιχειρήσεις κλπ , που αποστολή έχει την συνεχή επιτήρηση του χώρου και διατήρηση του σε ασφαλή κατάσταση.

## **1.2 Δομή τεύχους διπλωματικής εργασίας**

Το παρόν τεύχος έχει την παρακάτω δομή:

Αρχικά, υπάρχει μια σύντομη περίληψη της διπλωματικής εργασίας, στην οποία παρουσιάζεται συνοπτικά τα κύρια σημεία της. Στην συνέχεια ακολουθεί ο πίνακας περιεχομένων και η ευρεία περίληψη της εργασίας.

### **Κεφάλαιο 1: Εισαγωγή**

Πρόκειται για το παρόν κεφάλαιο, στο οποίο παρουσιάζεται συνοπτικά το θέμα της εργασίας και οι φάσεις εκπόνησης της.

### **Κεφάλαιο 2: Εργαλεία Ανάπτυξης Εφαρμογής**

Σε αυτό το κεφάλαιο γίνεται μια αναφορά στα εργαλεία τα οποία επιλέχτηκαν για την ανάπτυξη και υλοποίηση αυτής της εφαρμογής .

### **Κεφάλαιο 3: Σταδία Ανάπτυξης Εφαρμογής**

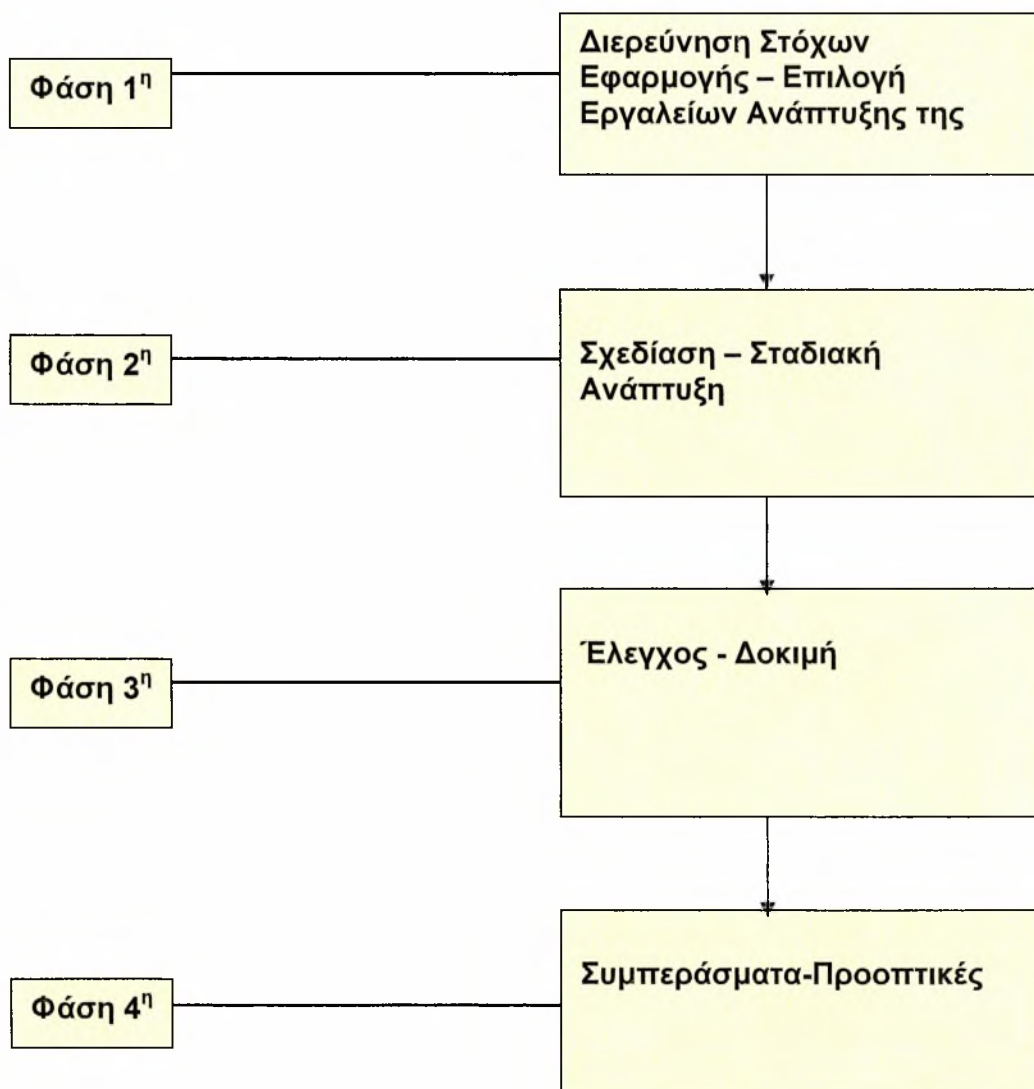
Το κεφάλαιο αυτό έχει ως σκοπό να αναλύσει τα στάδια ανάπτυξης της εφαρμογής, τις απαιτήσεις και τους στόχους που πρέπει να έχει, τον κατάλληλο σχεδιασμό και συγγραφή του κώδικα καθώς επίσης την δοκιμή και παρουσίασης της.

### **Κεφάλαιο 4: Συμπεράσματα- Προοπτικές**

Στο κεφάλαιο αυτό γίνεται μια σύνοψη των αποτελεσμάτων της διπλωματικής εργασίας και γίνονται κάποια σχόλια και παρατηρήσεις για τις προοπτικές της περιοχής.

### 1.3 Φάσεις της διπλωματικής εργασίας

Η εκπόνηση της διπλωματικής εργασίας πραγματοποιήθηκε μεταξύ Φεβρουαρίου 2008 και Οκτωβρίου 2008 και η πορεία αυτής ακολούθησε τις εξής φάσεις, που παρουσιάζονται παρακάτω στο Σχήμα 1.1.



**Σχήμα 1.1:** Φάσεις Διπλωματικής Εργασίας

## **Φάση 1η: Διερεύνηση στόχων Εφαρμογής – Επιλογή Εργαλείων Ανάπτυξης**

Στην πρώτη φάση, έγινε διερεύνηση των στόχων της εφαρμογής και κατόπιν διερεύνηση για την επιλογή κατάλληλων εργαλείων ανάπτυξης .

## **Φάση 2<sup>η</sup>: Σχεδίαση – Σταδιακή Ανάπτυξη**

Σε αυτή τη φάση έγινε η σχεδίαση της εφαρμογής πάνω στην οποία στηρίχτηκε η σταδιακή ανάπτυξη και η συγγραφή του κώδικα.

## **Φάση 3<sup>η</sup>: Έλεγχος - Δοκιμή**

Μετά την ολοκλήρωση όλων των σταδίων ανάπτυξης ακολούθησε ο έλεγχος για σωστή λειτουργία και τέλος η δόκιμη της εφαρμογής πάνω στο δίκτυο αισθητήρων που έγινε στο εργαστήριο .

## **Φάση 4<sup>η</sup>: Συμπεράσματα-Προοπτικές**

Σε αυτή τη φάση αναγνωρίστηκαν και καταγράφηκαν τα πιο σημαντικά σημεία από την προηγούμενη μελέτη και διερευνήθηκαν οι προοπτικές περαιτέρω μελέτης του συγκεκριμένου πεδίου.

---

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

### Εργαλεία Ανάπτυξης Εφαρμογής

---

## 2.1. Εισαγωγή

Το κύριο εργαλείο, ανάπτυξης της εφαρμογής, που επιλέχτηκε και χρησιμοποιήθηκε είναι το **NetBeans IDE 6.0** γραμμένο σε γλώσσα προγραμματισμού Java.

Για την εφαρμογή χρειάστηκε ένα σύστημα διαχείρισεως βάσεων δεδομένων και επιλέχτηκε και χρησιμοποιήθηκε το **MySQL Server 5.0** .

Τέλος , για τις ανάγκες διαχείρισης εφαρμογής πολυμέσων χρησιμοποιήθηκε το **Java Media Framework (JMF)**.

Στις επόμενες ενότητες αυτού του κεφαλαίου γίνεται εκτενέστερη αναφορά των εργαλείων αυτών.

## 2.2 NetBeans IDE 6.0

Το NetBeans IDE 6.0 είναι ένα έργο ανοικτού πηγαίου κώδικα και αναφέρεται, εκτός των άλλων, στην πλατφόρμα για ανάπτυξη Java εφαρμογών. Προσφέρει παρά πολλές δυνατότητες σε ένα προγραμματιστή που θέλει να αναπτύξει μια Java εφαρμογή ,με γραφικό περιβάλλον, και διασύνδεση με βάση δεδομένων .

## 2.3 MySQL Server 5.0

Η ανάγκη για διασύνδεση της εφαρμογής με μια βάση δεδομένων ώθησε στην επιλογή της εφαρμογής MySQL για την δημιουργία της βάσης δεδομένων που χρειάστηκε κατά την ανάπτυξη της εφαρμογής .

## 2.4 JavaMedia Framework (JMF)

Στους στόχους της εφαρμογής είναι και η διαχείριση οπτικοακουστικού υλικού, για το λόγο αυτό επιλέχτηκε και χρησιμοποιήθηκε το **Java Media Framework (JMF)**, βιβλιοθήκη της Java, το οποίο δίνει την δυνατότητα να ενσωματωθούν πολυμέσα σε Java εφαρμογές και να διαχειρίζονται μέσα από αυτές .

---

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>

### Στάδια Ανάπτυξης Εφαρμογής

---

### 3.1 Εξαγωγή και Ανάλυση Απαιτήσεων

Οι απαιτήσεις περιλαμβάνουν κυρίως τις ιδιότητες που έχει η υπό ανάπτυξη εφαρμογή.

Σύμφωνα με αυτές ,

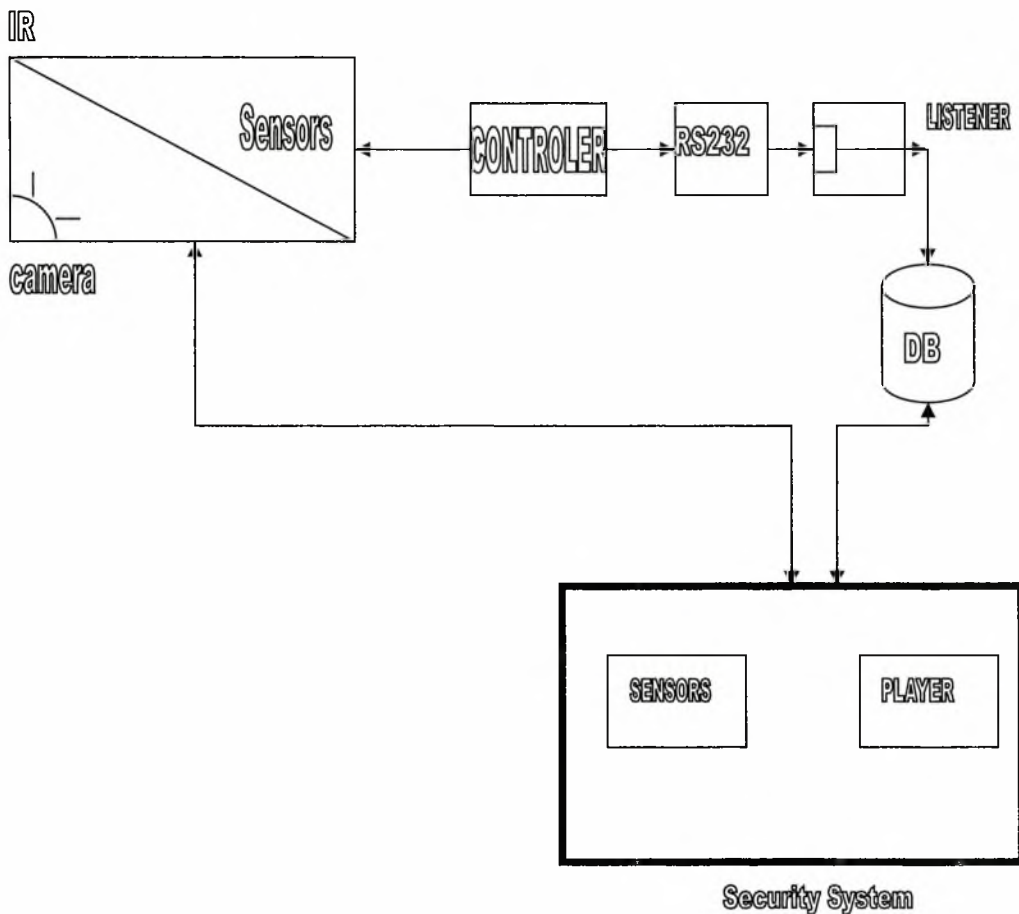
- η εφαρμογή θα πρέπει να αλληλεπιδρά με δίκτυα αισθητήρων και να επεξεργάζεται τα δεδομένα τα οποία αυτά συλλέγουν ,
- να κρατά ιστορικό του τύπου των αισθητήρων που χρησιμοποιούνται, των δεδομένων που συλλέγουν μέσα από την βάση με την οποία αλληλεπιδρά , το σύστημα και τα δίκτυα αισθητήρων ,
- να παρέχει την δυνατότητα στον χρήστη να ανατρέξει κατ' επιλογήν του σε συγκεκριμένα δεδομένα όποτε αυτά έχουν καταγραφεί και να εκτυπώσει τα αποτελέσματα,
- να τον ενημερώνει σε πραγματικό χρόνο με διάφορους τρόπους επικοινωνίας (μηνύματα, νέο αναδυόμενο παράθυρο, ηχητικό μήνυμα κ.α) , για κάποιο νέο συμβάν (alert) τη δεδομένη χρονική στιγμή,
- προτείνοντας του παράλληλα, ενέργειες που οφείλει αυτός να κάνει προς αντιμετώπιση του συμβάντος,
- εν συνεχεία να προσφέρει την δυνατότητα οπτικής επαφής του χώρου στον χρήστη , με χρήση μιας ή περισσότερων συσκευών απεικόνισης εικόνας (κάμερα),
- την λήψη εικόνας τη δεδομένη στιγμή που γίνεται κάποιο συμβάν (ήτοι αυξημένα επίπεδα θερμοκρασίας, υγρασίας, μη εξουσιοδοτημένη είσοδο στον χώρο),
- να αποθηκεύει την εικόνα, σε συγκεκριμένο αρχείο που έχει πρόσβαση ο χρήστης.
- Τέλος, να έχει απλό και εύχρηστο γραφικό περιβάλλον διασύνδεσης .



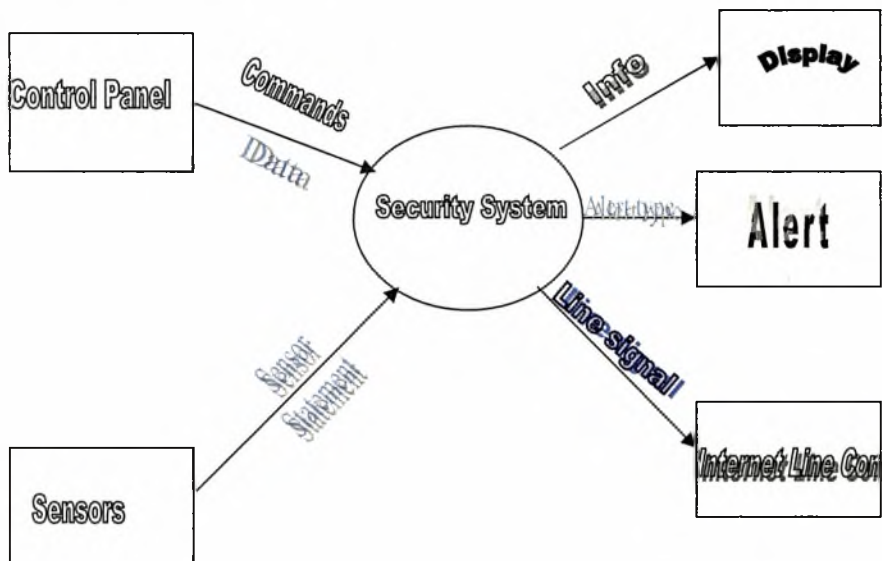
## 3.2 Σχεδίαση – Κωδικοποίηση

### 3.2.1 Σχεδίαση

Μετά από την εξαγωγή των απαιτήσεων ακολουθεί η σχεδίαση της εφαρμογής (σχ.3.1) καθώς και το διάγραμμα ροής (σχ.3.2), όπως αυτά φαίνονται παρακάτω .



Σχήμα 3.1: Σχεδίαση Εφαρμογής



**Σχήμα 3.2:** Διάγραμμα Ροής Δεδομένων

### 3.2.2 Κωδικοποίηση

Για την ανάπτυξη της εφαρμογής αυτής, η συγγραφή του κώδικα έγινε πάνω στο Java Desktop Application project με όνομα Sec Project, το οποίο βασίζεται στο Swing Application Framework (JSR 296) και παρέχεται από το NetBeans, με το οποίο μπορούμε να αναπτύξουμε εφαρμογές με γραφικό περιβάλλον, καθώς έχουμε και την δυνατότητα να αναπαραστήσουμε τους πίνακες μιας βάσης δεδομένων μέσα στην εφαρμογή, μια διεπαφή δηλαδή της Β.Δ σε γραφικό περιβάλλον.

Στην πορεία, δημιουργήθηκε μια βάση, αφού εγκαταστάθηκε πρώτα το MySQL Server 5.0, με όνομα HomeSecDatabase, username: root και password: danagno, δημιουργήθηκε σ' αυτήν ένας πίνακας δεδομένων με όνομα sensor\_info, στον οποίο περιέχονται τα δεδομένα που συλλέγουν οι αισθητήρες.

*Στιγμιότυπο κώδικα :*

```
create database HomeSecDatabase ;
```

```
use HomeSecDatabase ;
```

```
current database : HomeSecDatabase
```

Η βάση αυτή χρησιμοποιήθηκε και δοκιμάστηκε επιτυχώς σε όλη την πορεία της ανάπτυξης μέχρι που αντικαταστάθηκε με το sec\_table που περιέχει όλα τα απαραίτητα πεδία που χρειάζονται στην εφαρμογή, ώστε να συνδεθεί με το δίκτυο αισθητήρων και να λάβει σωστά τα δεδομένα.

Αυτά τα πεδία είναι :

- result\_time: η χρονική στιγμή που λάβαμε την πληροφορία.
- node\_id : το αναγνωριστικό(id) του κόμβου
- packet\_id : το αναγνωριστικό(id) του πακέτου
- temp : η τιμή της θερμοκρασίας που συλλέγει ο αισθητήρας
- humidity : η τιμή της υγρασίας που συλλέγει ο αισθητήρας
- ir : ανίχνευση κίνησης στο χώρο με τιμή ir ≠ 0
- int\_temp : η εσωτερική θερμοκρασία του αισθητήρα
- int\_voltage : τα volt της μπαταρίας του αισθητήρα

Στιγμιότυπο κώδικα :

```
CREATE TABLE IF NOT EXISTS `sec_table` (  
  `result_time` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  `nodeid` int(11) NOT NULL default '0',  
  `packetid` double NOT NULL default '0',  
  `ir` double default NULL,  
  `temp` double default NULL,  
  `humidity` double default NULL,  
  `int_voltage` double default NULL,  
  `int_temp` double default NULL,  
  PRIMARY KEY (`result_time`,`nodeid`,`packetid`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
mysql> describe sec_table;
```

Field	Type	Null	Key	Default	Extra
result_time	timestamp	NO	PRI	CURRENT_TIMESTAMP	
nodeid	int(11)	NO	PRI	0	
packetid	double	NO	PRI	0	
ir	double	YES		NULL	
temp	double	YES		NULL	
humidity	double	YES		NULL	
int_voltage	double	YES		NULL	
int_temp	double	YES		NULL	

Για την απεικόνιση της βάσης στην εφαρμογή , δημιουργήθηκε μια κλάση που κληρονομεί την `AbstractTableModel` , όπου ονόμασα `SensorTableModel` , και σαν είσοδο παίρνει τα δεδομένα από την μέθοδο `getResultFromSensors()` που δημιουργήθηκε για το σκοπό αυτό.

Στιγμιότυπο κώδικα :

```
public class SensorTableModel extends AbstractTableModel {  
  
    private java.sql.Statement stmt;  
  
    private int colnum=8;  
    private int rownum;  
    private String[] colNames={"ResultTime", "NodeID", "PacketID", "  
    private String[] headers;  
    private ArrayList<String[]> ResultSets;  
    private ResultSet rs;  
  
    public SensorTableModel(ResultSet rs) { ... }
```

```
public ResultSet getResultFromSensors() {  
  
    ResultSet rs=null;  
    try(  
        rs=stmt.executeQuery("Select * from sec_table");  
    )  
  
    catch(SQLException e){}  
  
    return rs;  
}
```

Για την σύνδεση της εφαρμογής στην βάση δεδομένων , δημιουργήθηκε μια κλάση, η MyDBConnection .

Στιγμιότυπο κώδικα :

```
public class MyDBConnection {  
  
    private MyDBConnection mdbc;  
    private java.sql.Statement stmt;  
    private Connection myConnection;  
  
    public MyDBConnection() {...}  
public void init() {...}  
  
    public Connection getMyConnection() {...}  
  
    public void close(ResultSet rs) {...}  
  
    public void close(java.sql.Statement stmt) {...}  
  
public void destroy() {...}  
}
```

Το Sec project αποτελείται από τρεις κλάσεις , μια για την σύνδεση στη ΒΔ που προαναφέραμε , μια για την διαχείριση της κάμερας που αλληλεπιδρά με την εφαρμογή , την SecCapture και μια κλάση που περιέχει μέσα και την main() , την secsec . Μαζί όλα αυτά περιέχονται στο java source package sec.

Στιγμιότυπο κώδικα :

```
public class secsec extends javax.swing.JFrame {  
    private boolean all=false;  
    boolean r;  
    AudioClip music = null;  
  
    public secsec() throws SQLException {  
  
        .....  
  
    }  
    .....  
    .....
```

Βασικό πρόβλημα στην πορεία της ανάπτυξης ήταν η ενημέρωση του `SensorTableModel` από την `HomeSecDatabase` σε πραγματικό χρόνο, αφού πρόκειται για μια εξωτερική βάση δεδομένων, χωρίς να γίνεται κάθε φορά εκκίνηση της εφαρμογής.

Το εμπόδιο αυτό ξεπεράστηκε προσθέτοντας στο `secsec` μια κλάση από `timers` (`Reaptimer1`), αναγκάζοντας έτσι το `SensorTableModel` να δημιουργεί αυτόματα εκ νέου το δομητή `SensorTableModel()` ανά τακτά χρονικά διαστήματα που έχουν οριστεί, ώστε να έχει συνεχώς ενημερωμένα τα δεδομένα.

Στιγμιότυπο κώδικα :

```
public class Reaptimer1 {  
  
    private final Timer timer = new Timer();  
    private final int minutes;  
    private final int period = 15000;  
    public Reaptimer1(int minutes) {...}  
  
    public void start() {  
        timer.schedule(new TimerTask() {  
            public void run() {...}  
            private void refresh() {...}  
        }, minutes * 6 * 1000, period);  
    }  
}
```



Για τις ανάγκες της εφαρμογής χρειάστηκε να δημιουργηθούν έξι (6) μοντέλα `SensorTableModel` τα οποία είναι :

1. **jTable1**: Το οποίο εμφανίζει τα δεδομένα που συλλέγουν οι αισθητήρες τις τελευταίες έξι ώρες.
2. **jTable2**: Το οποίο εμφανίζει το ιστορικό όλων των δεδομένων που συλλέγουν οι αισθητήρες.
3. **jTable3** : Το οποίο εμφανίζει σε νέο αναδυόμενο παράθυρο (**jFrame1**) κάποιο νέο συμβάν ανίχνευσης κίνησης (`alert`).
4. **jTable4**: Το οποίο εμφανίζει σε νέο αναδυόμενο παράθυρο (**jFrame2**) κάποιο νέο συμβάν υψηλών επιπέδων θερμοκρασίας (`alert`).
5. **jTable5**: Το οποίο εμφανίζει σε νέο αναδυόμενο παράθυρο (**jFrame3**) κάποιο νέο συμβάν υψηλών επιπέδων υγρασίας (`alert`).
6. **jTable6**: Το οποίο επιστρέφει σε νέο αναδυόμενο παράθυρο (**jFrame4**) τα αποτελέσματα από την αναζήτηση που κάνει ο χρήστης για κάποια δεδομένα που τον ενδιαφέρουν από την βάση δεδομένων.

Όσον αφορά την διαχείριση των πολυμέσων , έχοντας εγκαταστήσει το Java Media Framework (JMF) , στην συγκεκριμένη περίπτωση την έκδοση 2.1.1e, στη κλάση `SecCapture` που δημιουργήθηκε για το σκοπό αυτό , γράφτηκε κατάλληλος κώδικας ώστε να ανιχνεύει την κάμερα ή τις κάμερες που είναι εγκατεστημένες στον υπολογιστή που τρέχει η εφαρμογή , και να τη διαχειρίζεται μέσω του `JInternalFrame1` που προστέθηκε ώστε να φαίνεται ένα είδος player με κατάλληλα κουμπιά on/off και capture .

Στιγμιότυπο κώδικα :

```
public class SecCapture extends Panel implements ActionListener
{
    public SecCapture()
    {
        (...)
    }

    public static void playerclose()
    {
        (...)
    }
}
```

Λόγω των απαιτήσεων της εφαρμογής δεν χρειάστηκε κώδικας για την καταγραφή video παρά μόνο για την λήψη εικόνας . Όμως είναι μια καλή ιδέα για μετέπειτα επέκταση του , σε προδιαγραφές που μπορεί να ζητήσει κάποιος .

Επίσης , κατά την εμφάνιση ενός νέου συμβάντος (alert) , η εφαρμογή αυτόματα καταγράφει την εικόνα του χώρου και την αποθηκεύει σε ένα φάκελο στο σκληρό δίσκο , η διαδρομή του οποίου έχει προκαθοριστεί κατά την συγγραφή του κώδικα . Σίγουρα αυτός ο φάκελος πρέπει να δημιουργεί από τον χρήστη , γιατί δεν παρέχεται η δυνατότητα να δημιουργηθεί από την εφαρμογή αυτόματα. Και αυτό ίσως θα μπορούσε να προστεθεί σε μια μελλοντική επέκταση της εφαρμογής .

Ο φάκελος αυτός έχει το όνομα `Hs_Cam_Data` και βρίσκεται στο :

**C:\Hs\_Cam\_Data**



Ένα ακόμη ενδιαφέρον κομμάτι στον κώδικα , είναι η επιλογή που δίνει στο χρήστη, να αναζητήσει οποιαδήποτε δεδομένα από την βάση .  
Για το σκοπό αυτό δημιουργήθηκε ένα επιπλέον αναδυόμενο παράθυρο (**jFrame4**) το οποίο εμφανίζεται πατώντας το πλήκτρο που φαίνεται στο βασικό μενού της εφαρμογής(**jButton3**) (Search\_Query), μάλιστα μπορεί να το επιλέξει κανείς με την συντόμευση που έχει δημιουργηθεί , όπου είναι ο συνδυασμός πλήκτρων ( Ctrl+Alt+Q ) .

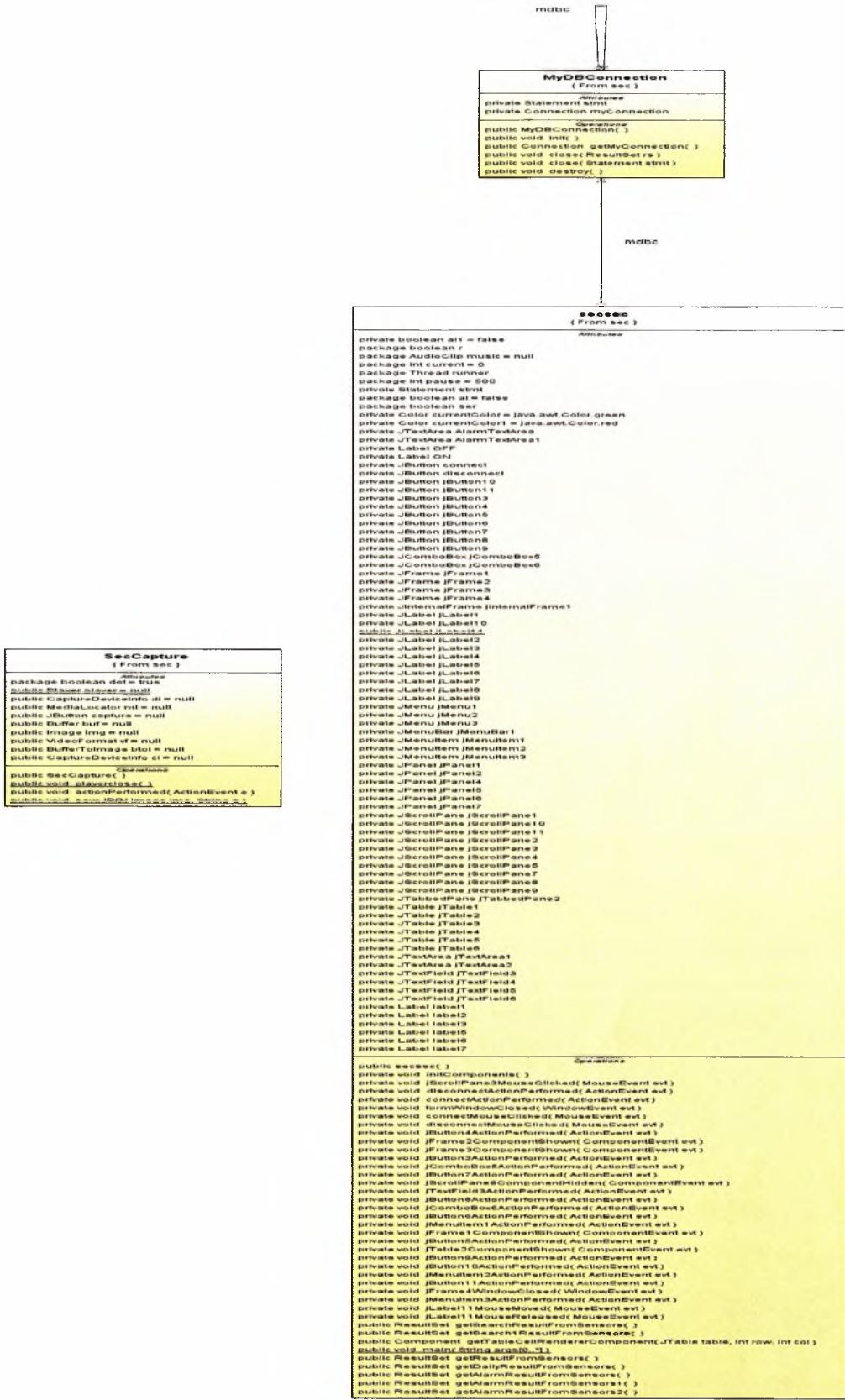
Στιγμιότυπο κώδικα :

```
private void jMenuItemActionPerformed(java.awt.event.ActionEvent evt) {  
  
    JFrame4.setVisible(true) ;  
  
}
```

Στο νέο λοιπόν, αυτό παράθυρο , έχει δημιουργηθεί μια φόρμα για εισαγωγή των δεδομένων προς αναζήτηση και έχει επίσης δημιουργηθεί και μια περιοχή που δείχνει κατά την αναζήτηση το **Query Builder** .

```
jTextArea1.setText("Select * from sec_table where \n"+"nodeid="+sname0+  
    "\n and"+" \n"+"result_time > " + "'" +stime2+"'" +  
    "\n and"+" \n"+"result_time < " + "'" +stime1+"'" +  
    "\n and"+" \n"+"sname1+svalue1+svalue2+ ";" ) ;  
str = jTextArea1.getText();
```

Στην επόμενη σελίδα φαίνεται ένα διάγραμμα UML του κώδικα που σχολιάζουμε , στο οποίο αναπαρίστανται οι κλάσεις secsec, SecCapture και η MySqlConnection .



Σχήμα 3.3: Διάγραμμα UML

### 3.3 Δοκιμή της Εφαρμογής

Η εφαρμογή δοκιμάστηκε αρχικά και σε όλα τα στάδια ανάπτυξης της με έτοιμα δεδομένα στην βάση και ανταποκρίθηκε επαρκώς στις απαιτήσεις για τις οποίες σχεδιάστηκε.

Είναι πλέον έτοιμη η εφαρμογή , για το επόμενο κρίσιμο στάδιο , το οποίο είναι η διασύνδεση της με το δίκτυο αισθητήρων .

Στο στάδιο αυτό , θα πρέπει να αναμένουμε , να παίρνει σαν είσοδο τα δεδομένα που περισυλλέγουν οι αισθητήρες και στέλνουν στην βάση , και να τα εμφανίζει .

Οι Δοκιμές και η καταγραφή αποτελεσμάτων έγιναν στο εργαστήριο αισθητήρων .

Χρησιμοποιήθηκε η πλατφόρμα Tmote Sky , η οποία είναι μια ασύρματη μονάδα που χρησιμοποιείται για ασύρματα δίκτυα αισθητήρων και βασίζεται στα εγγυοστασιακά πρότυπα USB και IEEE 802.15.4

Χαρακτηριστικά :

- 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver
- Interoperability with other IEEE 802.15.4 devices
- 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash)
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller
- Integrated onboard antenna with 50m range indoors / 125m range outdoors
- Integrated Humidity, Temperature, and Light sensors
- Ultra low current consumption
- Fast wakeup from sleep (<6μs)
- Hardware link-layer encryption and authentication
- Programming and data collection via USB
- 16-pin expansion support and optional SMA antenna connector
- TinyOS support : mesh networking and communication implementation
- Complies with FCC Part 15 and Industry Canada regulations
- Environmentally friendly – complies with RoHS regulations



**Tmote Sky**

Οι αισθητήρες θερμοκρασίας και υγρασίας είναι οι SHT11 και SHT15 και τα χαρακτηριστικά τους :

Parameter	MIN	TYP	MAX	Units
<b>Humidity</b>				
Resolution	0.5	0.03	0.03	%RH
	8	12	12	Bit
Repeatability		±0.1		%RH
Range	0		100	%RH
<b>Temperature</b>				
Resolution	0.04	0.01	0.01	°C
	0.07	0.02	0.02	°F
	12	14	14	bit
Repeatability		±0.1		°C
		±0.2		°F
Range	-40		123.8	°C
	-40		254.9	°F

#### Sensirion relative humidity and temperature performance specifications

Επίσης για την ανίχνευση κίνησης χρησιμοποιήθηκε ο αισθητήρας WIEye της EasySen ο οποίος χρησιμοποιείται για ασύρματη παρακολούθηση και είναι ιδανικός για τον σκοπό της εφαρμογής

Χαρακτηριστικά :

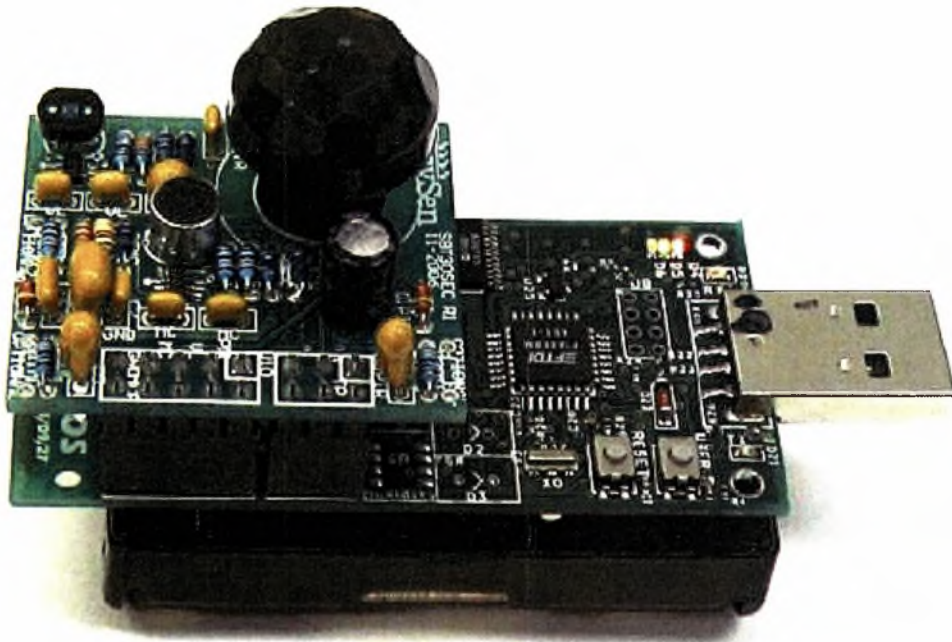
Long-range passive infrared (PIR) sensor well suited for detecting human presence or vehicles

(90-100° wide detection cone, 20-30 feet detection range for human presence, 50-150 feet

detection range for vehicles depending on the size of vehicle)

- Visual light and acoustic sensors complementing the PIR sensor for improved detection
- Adjustable time constant in PIR sensor output for customized applications
- Rectified acoustic envelope output with adjustable time constant and adjustable low-pass filter for customized applications
- Additional acoustic channel allowing high frequency sampling of voice and other noise signals
- Can be directly plugged into the external connector of IEEE 802.15.4/Zigbee compliant TelosB motes
- Smaller form factor than TelosB motes
- TinyOS and Java code support available for sampling sensor channels, display readings on a PC
- Power saving mode

Στην επομενη σελιδα φαίνεται εικόνα του αισθητήρα



## WiEye

Η εφαρμογή δοκιμάστηκε χωρίς να παρουσιάσει προβλήματα και ανταποκρίθηκε επιτυχώς .

Κάποια από τα αποτελέσματα φαίνονται παρακάτω

ResultTime	Nodeid	PacketID	IR	Temp	Humidity	SensorVolt	InSensorTe
2008-10-08 17:20:00.0	5	531	0	19	45	1.139648438	2883
2008-10-09 08:31:00.0	5	570	0	36	20	1.139648437	2882
2008-10-15 13:48:53.0	5	590	0	56	20	1.139648437	2882
2008-10-15 14:01:11.0	5	596	0	37	90	1.139648437	2882
2008-10-15 14:03:40.0	5	596	0	37	96	1.139648437	2882
2008-10-15 14:10:19.0	5	596	0	57	73	1.139648437	2882
2008-10-15 14:11:04.0	5	596	0	57	73	1.139648437	2882
2008-10-15 14:12:11.0	5	600	0	67	73	1.139648437	2882
2008-10-15 14:17:43.0	5	606	0	27	87	1.139648437	2882
2008-10-15 14:19:30.0	5	608	0	27	92	1.139648437	2882
2008-10-15 14:23:09.0	5	610	4000	27	52	1.139648437	2882
2008-10-15 14:24:58.0	5	612	0	42	52	1.139648437	2882
2008-10-15 16:04:30.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:05:15.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:18:47.0	5	612	4000	22	31	1.139648437	2882

## Αποτελέσματα

## 3.4 Παρουσίαση της Εφαρμογής

### 3.4.1 Ενέργειες πριν την Εκκίνηση

Η εφαρμογή για να διανεμηθεί και να λειτουργήσει σωστά , πριν την εκκίνηση της , θα πρέπει :

- Να έχει ήδη εγκατασταθεί το Java(TM) Platform, Standard Edition Runtime Environment (JRE) Version 6 Java Development Kit (JDK1.6)
- Επίσης να έχει εγκατασταθεί ο MySQL Server 5.0 και να έχει δημιουργεί ήδη η βάση που προαναφέρεται στο προηγούμενο κεφάλαιο
- Να έχει εγκατασταθεί το JMF
- Screen resolution 1280 by 720 pixels.

### 3.4.2 Τρόποι εκκίνησης

Έχοντας λοιπόν εγκατασταθεί τα απαραίτητα προγράμματα , είναι πλέον έτοιμη η εφαρμογή να ξεκινήσει.

Οι τρόποι εκκίνησης της εφαρμογής είναι οι εξής :

1. Διπλό κλικ πάνω στο αρχείο Sec.jar , και ανοίγει απευθείας η εφαρμογή
2. Διπλό κλικ πάνω στο αρχείο launch.jsp
3. Και τέλος από το Command Prompt με την εντολή

```
java -jar "Sec.jar"
```

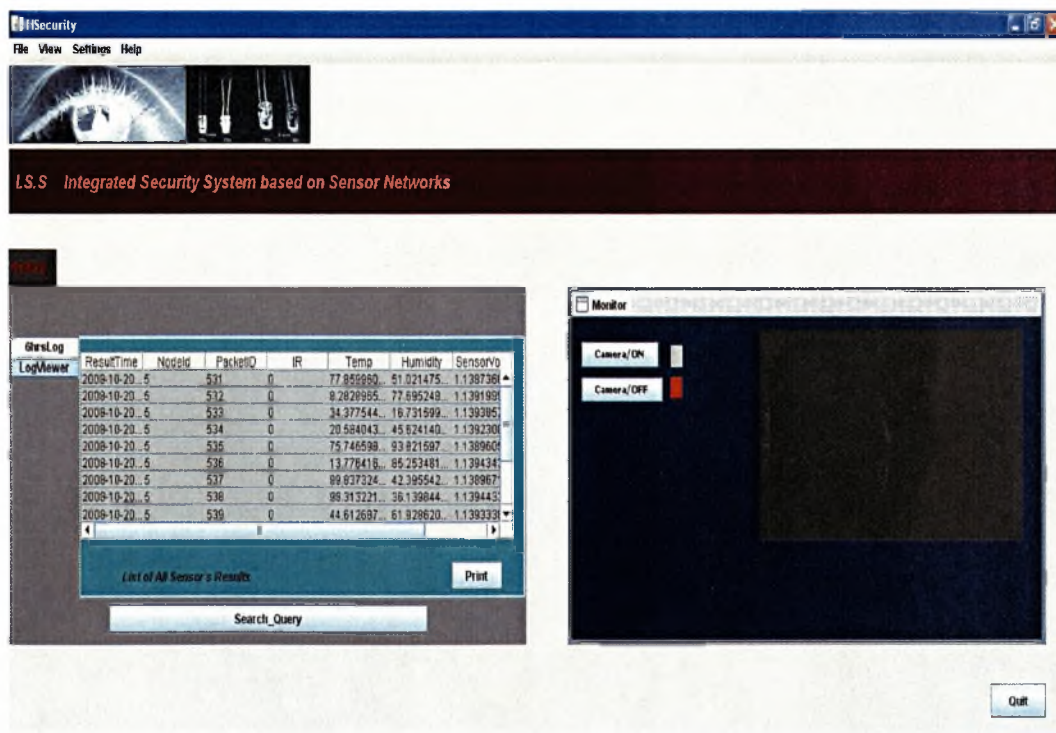
### 3.4.3 Παρουσίαση Γραφικού Περιβάλλοντος

Στην ενότητα αυτή γίνεται παρουσίαση του γραφικού περιβάλλοντος της εφαρμογής ,

Ξεκινώντας την εφαρμογή με τους τρόπους που αναφέρθηκαν στην προηγούμενη ενότητα , συναντάμε το βασικό μενού της εφαρμογής το οποίο αποτελείται από 3 πεδία ,

1. Το πρώτο περιέχει όλες τις πληροφορίες από την βάση δεδομένων των αισθητήρων
2. Το δεύτερο περιέχει το μόνιτορ απ' όπου γίνεται η παρακολούθηση του χώρου
3. Και τελευταίο , την μπάρα με τα μενού επιλογών .

Στην παρακάτω εικόνα 3.1 , φαίνεται το αρχικό μενού της εφαρμογής κατά την εκκίνηση της .



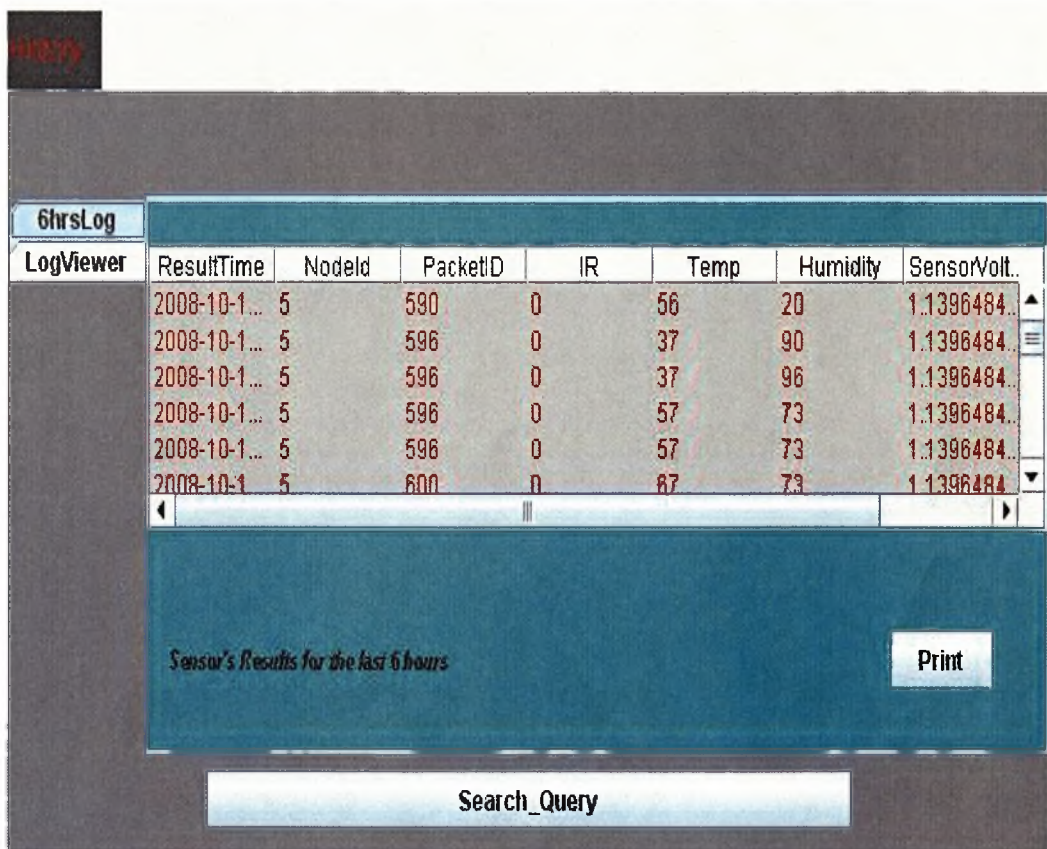
Βασικό Μενού Εφαρμογής - Εικόνα 3.1

1. Στο πρώτο πεδίο που αναφέραμε προηγουμένως, μπορούμε να πάρουμε όλες τις απαραίτητες πληροφορίες που συλλέγουν οι αισθητήρες οποιαδήποτε στιγμή θέλουμε .

Τα δεδομένα αυτά είναι τα περιεχόμενα του πίνακα που φαίνονται στο πεδίο.

- Στον πρώτο πίνακα **6hrsLog** του πεδίου, δίνονται οι πληροφορίες των αισθητήρων που έχουν συλλεχτεί τις τελευταίες 6 ώρες . Επίσης μας δίνεται η δυνατότητα να εκτυπώσουμε τα δεδομένα αυτά για περαιτέρω ανάλυση κάνοντας ένα κλικ στο κουμπί **Print** .

Στη συνέχεια παρατίθεται ένα δείγμα του πίνακα αυτού (Εικόνα 3.2).



The screenshot shows a web interface titled "6hrsLog" with a "LogViewer" section. It contains a table with the following data:

ResultTime	NodeID	PacketID	IR	Temp	Humidity	SensorVolt..
2008-10-1...	5	590	0	56	20	1.1396484..
2008-10-1...	5	596	0	37	90	1.1396484..
2008-10-1...	5	596	0	37	96	1.1396484..
2008-10-1...	5	596	0	57	73	1.1396484..
2008-10-1...	5	596	0	57	73	1.1396484..
2008-10-1...	5	600	0	67	73	1.1396484..

Below the table, there is a "Print" button and a "Search\_Query" input field. The text "Sensor's Results for the last 6 hours" is displayed above the button.

6hrsLog - Εικόνα 3.2



- Μπορούμε επίσης να δούμε όλο το ιστορικό των δεδομένων επιλέγοντας την ετικέτα **LogViewer** όπου εμφανίζεται ένας δεύτερος πίνακας στην θέση που ήταν ο πρώτος .

Και εδώ, όπως και πριν, έχουμε την επιλογή να εκτυπώσουμε τα δεδομένα κάνοντας κλικ στο κουμπί **Print** .

Παρακάτω φαίνεται ένα δείγμα του ιστορικού των δεδομένων (Εικόνα 3.3), καθώς και τα αποτελέσματα από την επιλογή Print (Εικόνα 3.4).

ResultTime	NodeId	PacketID	IR	Temp	Humidity
2008-10-15 14:01:11.0	5	596	0	37	90
2008-10-15 14:03:40.0	5	596	0	37	96
2008-10-15 14:10:19.0	5	596	0	57	73
2008-10-15 14:11:04.0	5	596	0	57	73
2008-10-15 14:12:11.0	5	600	0	67	73
2008-10-15 14:17:43.0	5	606	0	27	87
2008-10-15 14:19:30.0	5	608	0	27	92
2008-10-15 14:23:09.0	5	610	4000	27	52
2008-10-15 14:24:58.0	5	612	0	42	52

*List of All Sensor's Results* **Print**

Search\_Query

**LogViewer - Εικόνα 3.3**

## Page 1 All Sensors Results

ResultTime	Nodeld	PacketID	IR	Temp	Humidity	SensorVolt	InSensorTe
2008-04-22 12:16:00.0	5	526	0	20	40	1.139648438	2883
2008-04-22 12:16:00.0	5	527	0	15	67	1.139648438	2883
2008-04-22 12:16:00.0	5	529	0	33	49	1.139648438	2883
2008-04-22 12:16:00.0	5	530	0	25	14	1.139648438	2883
2008-10-08 17:20:00.0	5	531	0	19	45	1.139648438	2883
2008-10-09 08:31:00.0	5	570	0	36	20	1.139648437	2882
2008-10-15 13:48:53.0	5	590	0	56	20	1.139648437	2882
2008-10-15 14:01:11.0	5	596	0	37	90	1.139648437	2882
2008-10-15 14:03:40.0	5	596	0	37	96	1.139648437	2882
2008-10-15 14:10:19.0	5	596	0	57	73	1.139648437	2882
2008-10-15 14:11:04.0	5	596	0	57	73	1.139648437	2882
2008-10-15 14:12:11.0	5	600	0	67	73	1.139648437	2882
2008-10-15 14:17:43.0	5	606	0	27	87	1.139648437	2882
2008-10-15 14:19:30.0	5	608	0	27	92	1.139648437	2882
2008-10-15 14:23:09.0	5	610	4000	27	52	1.139648437	2882
2008-10-15 14:24:58.0	5	612	0	42	52	1.139648437	2882
2008-10-15 16:04:30.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:05:15.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:18:47.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:19:36.0	5	612	0	122	31	1.139648437	2882
2008-10-15 16:23:18.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:24:40.0	5	612	4000	22	31	1.139648437	2882
2008-10-15 16:25:23.0	5	612	4000	22	31	1.139648437	2882

### Print - Εικόνα 3.4

- Μια ακόμη δυνατότητα που μας δίνεται στο πρώτο πεδίο της εφαρμογής, είναι να κάνουμε όποια αναζήτηση θέλουμε στην βάση δεδομένων .

Αυτό γίνεται κάνοντας κλικ στο κουμπί **Search\_Query** , όπου μας οδηγεί σε ένα νέο αναδυόμενο παράθυρο.

Το παράθυρο αυτό αποτελείται από

- μια φόρμα συμπλήρωσης των στοιχείων που θέλουμε να αναζητήσουμε ,
- μια περιοχή κειμένου (**Query Builder**) στην οποία εμφανίζεται η εντολή SQL που παράγεται από την εισαγωγή των τιμών στη φόρμα συμπλήρωσης
- και τέλος τον πίνακα με τα αποτελέσματα της αναζήτησης .

Επίσης και σε αυτή την περίπτωση μπορούμε να εκτυπώσουμε τα αποτελέσματα της αναζήτησής μας κάνοντας κλικ στο κουμπί **Print**.

Σε περίπτωση εισαγωγής λανθασμένων δεδομένων ή διαφορετικού τύπου δεδομένων από αυτά που ορίζει η φόρμα , έχουμε την δυνατότητα να καθαρίσουμε την φόρμα , απλά κάνοντας κλικ στο κουμπί **Clear** και να συνεχίσουμε με μια νέα αναζήτηση.

Παρακάτω δίνεται η εικόνα του αναδυόμενου παραθύρου (Searching Query Form - Εικόνα 3.5) .

**Searching Query Form**

**Date**

Node Id: 5    Temp/Hum/IR: temp    From: 2008-04-21    To: 2008-04-24    Value: 15    Run    Clear

**Query Builder**

```
Select * from sec_table where
nodeid=5 and
result_time > '2008-04-21' and
result_time < '2008-04-24' and
temp>15;
```

**Results**

ResultTime	NodeId	PacketID	IR	Temp	Humidity	SensorVol...	InSensorTe...
2008-04-22 12:16:00.0	5	526	0	20	40	1.139648...	2883
2008-04-22 12:16:00.0	5	529	0	33	49	1.139648...	2883
2008-04-22 12:16:00.0	5	530	0	25	14	1.139648...	2883

Print    Exit

**Searching Query Form - Εικόνα 3.5**

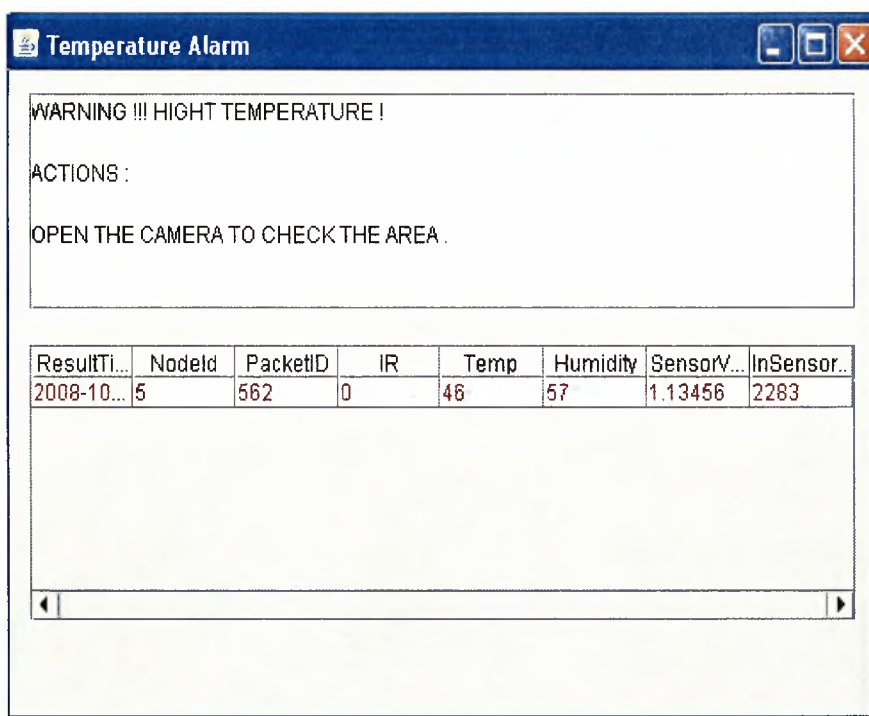
- Στο πεδίο αυτό, των πληροφοριών δηλαδή που μας προσφέρει η εφαρμογή από τη βάση δεδομένων των αισθητήρων μπορούμε να αναφέρουμε και την κατηγορία των alerts που έρχονται από τους αισθητήρες

Αυτά είναι τα εξής :

- **Temperature Alarm** , όπου εμφανίζει ένα νέο αναδυόμενο παράθυρο , συνοδευόμενο με ηχητικό μήνυμα (σειρήνα) και αποτελείται από μια περιοχή κειμένου , στην οποία αναγράφεται το είδος του alert, στη συγκεκριμένη περίπτωση , αυξημένα επίπεδα θερμοκρασίας στο χώρο , απο τις προτεινόμενες ενέργειες, και από τον πίνακα με την πληροφορία του αισθητήρα. Επίσης τη στιγμή που ενεργοποιείται ένα alert , αυτόματα γίνεται μια λήψη του χώρου από την κάμερα η οποία αποθηκεύεται στο c:Hs\_Data\_Cam σε μορφή jpeg και όνομα την ώρα και ημερομηνία του συμβάντος ,απ' όπου ο χρήστης μπορεί να ανατρέξει και να την επεξεργαστεί .

Σε αυτό το σημείο αξίζει να σημειωθεί ότι η πληροφορία και γενικά το αναδυόμενο αυτό παράθυρο είναι μόνο για να ενημερώσει εγκαίρως το χρήστη για το συμβάν (εκτιμώμενος χρόνος περίπου 3 sec) και όχι για περαιτέρω επεξεργασία . Αν θέλει ο χρήστης να δει αργότερα την πληροφορία αυτή μπορεί να την αναζητήσει από τον **6hrsLog** πίνακα δεδομένων .

Στην επόμενη σελίδα δίνεται στιγμιότυπο του αναδυόμενου παραθύρου Temperature Alarm – Εικόνα 3.6 .



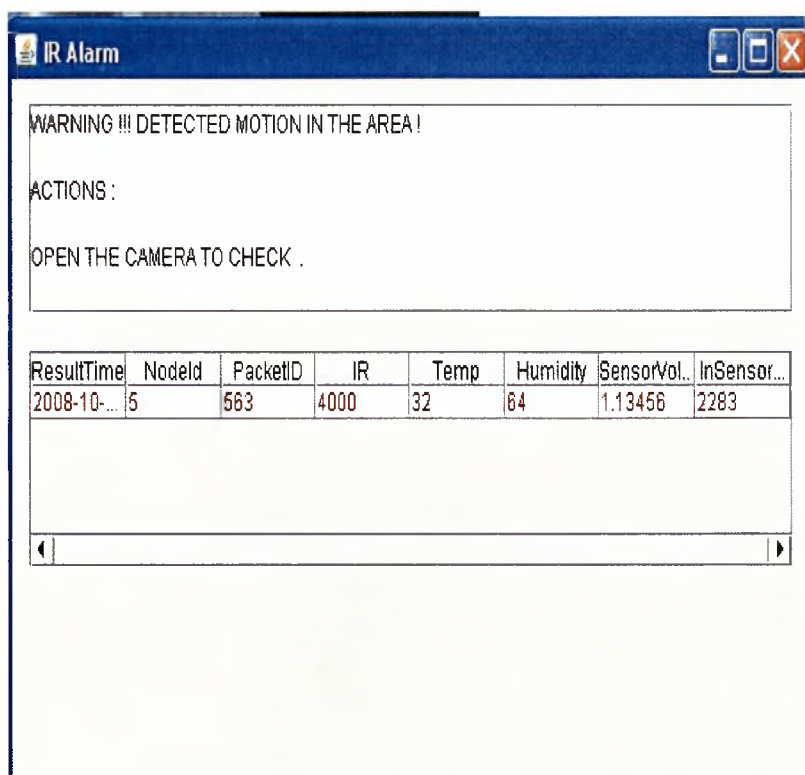
**Temperature Alarm – Εικόνα 3.6**

- **Humidity Alarm** , όπου εμφανίζει ένα νέο αναδυόμενο παράθυρο , συνοδευόμενο με ηχητικό μήνυμα (σειρήνα) και αποτελείται όπως και στην περίπτωση του alert που περιγράψαμε προηγουμένως , από μια περιοχή κειμένου , στην οποία αναγράφεται το είδος του alert, στη συγκεκριμένη περίπτωση , αυξημένα επίπεδα υγρασίας στο χώρο, και προτεινόμενες ενέργειες, καθώς επίσης και από τον πίνακα με την πληροφορία του αισθητήρα. Στην επόμενη σελίδα δίνεται στιγμιότυπο του αναδυόμενου παραθύρου Humidity Alarm – Εικόνα 3.7 .

WARNING !!! HIGHT HUMIDITY !							
ACTIONS :							
OPEN CAMERA TO CHECK THE AREA .							
ResultTi...	Nodeld	PacketID	IR	Temp	Humidity	SensorVo...	InSensor...
2008-10-...	5	612	0	22	91	1.13964...	2882

**Humidity Alarm – Εικόνα 3.7**

- **IR Alarm** , όπου εμφανίζει ένα νέο αναδυόμενο παράθυρο, συνοδευόμενο με ηχητικό μήνυμα (σειρήνα) και αποτελείται όπως και στην περίπτωση του alert που περιγράψαμε προηγουμένως, από μια περιοχή κειμένου, στην οποία αναγράφεται το είδος του alert, στη συγκεκριμένη περίπτωση, ανίχνευση κίνησης στο χώρο, και προτεινόμενες ενέργειες καθώς επίσης και από τον πίνακα με την πληροφορία του αισθητήρα.  
Στην επόμενη σελίδα δίνεται στιγμιότυπο του αναδυόμενου παραθύρου IR Alarm – Εικόνα 3.8



IR Alarm – Εικόνα 3.8

2. Στο δεύτερο πεδίο, περιέχεται το μόνιτορ, απ' το οποίο μπορούμε να έχουμε οπτική επαφή στο χώρο .

Αυτό αποτελείται από τρία κουμπιά

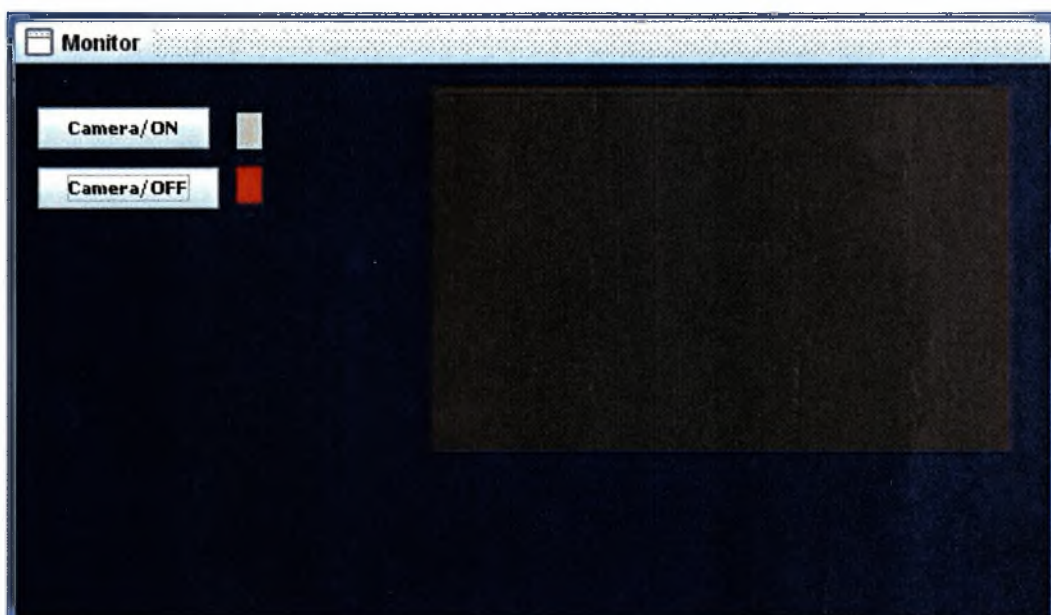
- ένα για την ενεργοποίηση της κάμερας , Camera / On , στο οποίο, κάνοντας κλικ μια φορά, εμφανίζεται στο μόνιτορ ο χώρος στον οποίο έχει εγκατασταθεί η κάμερα για την παρακολούθηση του .
- ένα για την απενεργοποίηση της κάμερας, Camera /Off , στο οποίο, κάνοντας κλικ μια φορά, μπαίνει η κάμερα σε αναμονή και δεν φαίνεται ο χώρος στο μόνιτορ,

Στην επόμενη σελίδα δίνεται στιγμιότυπο του μόνιτορ στη φάση ενεργοποίησης (Εικόνα 3.9), στην φάση απενεργοποίησης(Εικόνα 3.10)





Camera / On – Εικόνα 3.9



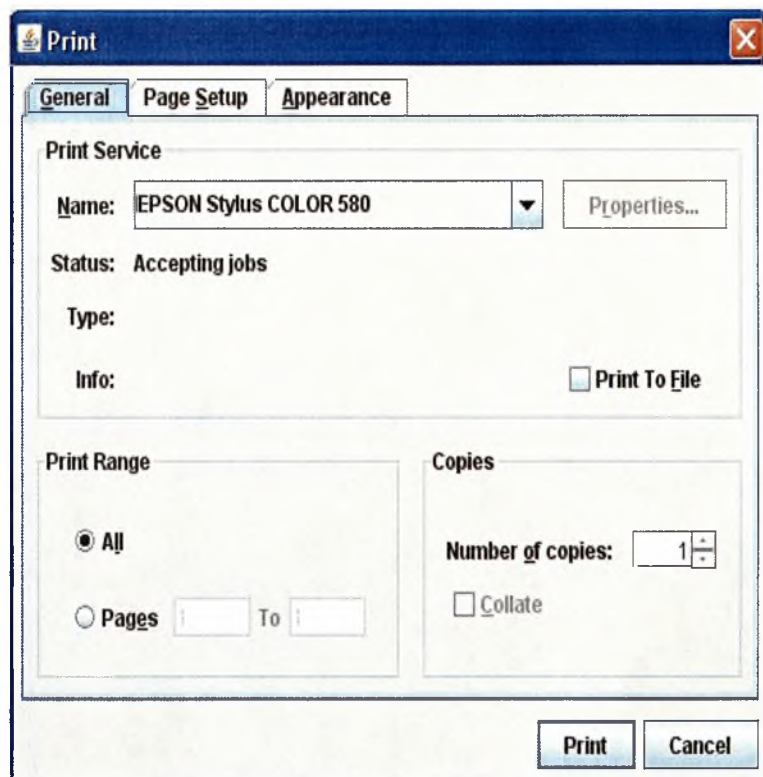
Camera / Off – Εικόνα 3.10

3. Στο τρίτο και τελευταίο πεδίο της εφαρμογής , βρίσκεται η μπάρα από το μενού τριών επιλογών , οι οποίες είναι

➤ **File** ,το οποίο αποτελείται από δυο στοιχεία ,

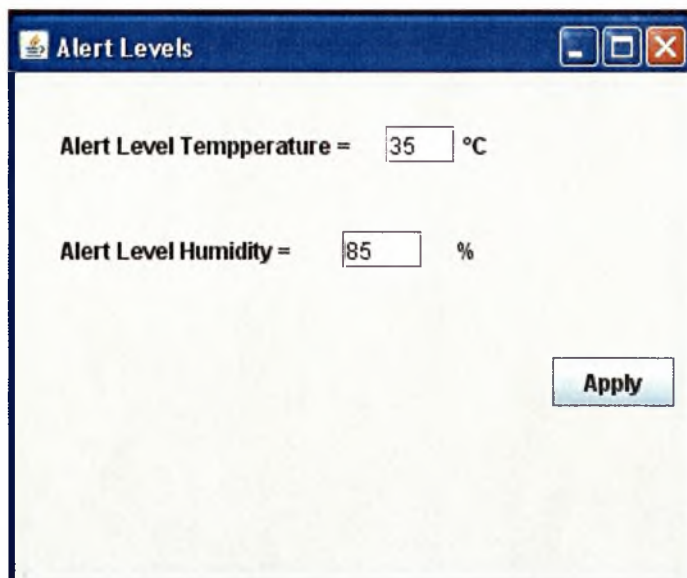
✚ Το **Search Query** , όπου μπορεί να ενεργοποιηθεί και με το συνδυασμό των πλήκτρων **Ctrl+Alt+Q**

✚ Και το **Print Sensor Results** , όπου μπορεί να ενεργοποιηθεί και με το συνδυασμό των πλήκτρων **Ctrl+Alt+P** , και να εκτυπώσει τα περιεχόμενα του πίνακα **LogViewer**



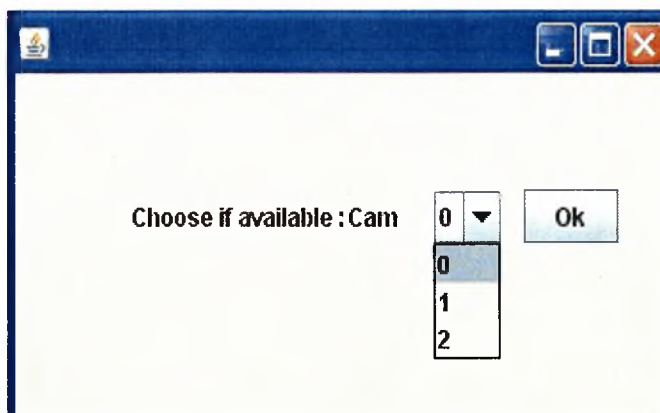
Print – Εικόνα 3.11

- **View** , το οποίο αποτελείται από δυο στοιχεία
  - ✚ Το Reload , όπου μπορεί να ενεργοποιηθεί και με το συνδυασμό των πλήκτρων **Cntrl+Alt+Q** και το οποίο κάνει ένα είδος ανανέωσης του κεντρικού παραθύρου της εφαρμογής .
  
- **Settings** , το οποίο αποτελείται προς το παρόν από
  - ✚ Το Set Sensor Level , όπου μπορεί ο χρήστης να αλλάξει εφόσον επιθυμεί και το απαιτούν κάθε φορά οι συνθήκες , τα επίπεδα των τιμών για τα οποία συμβαίνει κάποιο alert .



**Alert Levels – Εικόνα 3.12**

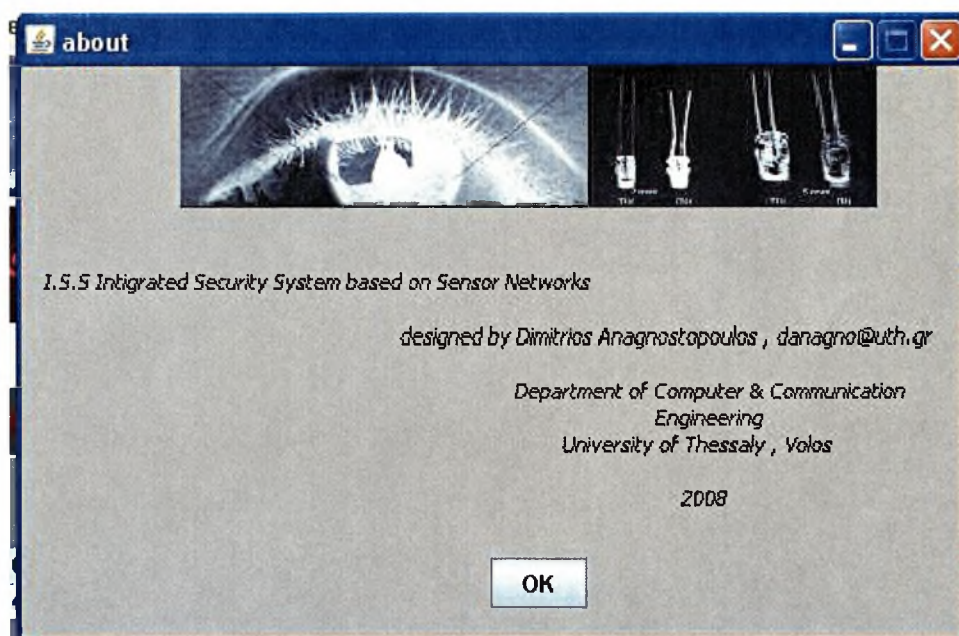
- ✚ Το Choose Cam Device , όπου μπορεί ο χρήστης να επιλέξει ποια κάμερα θα τεθεί σε εφαρμογή , στην περίπτωση που έχουμε περισσότερες από μια κάμερες τοποθετημένες στο χώρο.



Choose camera – Εικόνα 3.12

- **Help** , το οποίο αποτελείται προς το παρόν από

- ✚ Το About , όπου δείχνει τα , περί την εφαρμογή σχετικά .



About – Εικόνα 3.13

---

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>

### ΣΥΜΠΕΡΑΣΜΑΤΑ - ΠΡΟΟΠΤΙΚΕΣ

---

## 4.1 Συμπεράσματα – Προοπτικές

Ύστερα από διερεύνηση και επιλογή των εργαλείων ανάπτυξης, καθώς επίσης από την ανάλυση των απαιτήσεων και τον σχεδιασμό , αναπτύχθηκε το ολοκληρωμένο σύστημα ασφάλειας και δοκιμάστηκε στο εργαστήριο και είχε τα αποτελέσματα που αναμέναμε .

Οι προοπτικές που απορρέουν από το σύστημα αυτό έχουν ως εξής:

- Μελλοντική επέκταση της εφαρμογής, να εφαρμοστεί και να προσφέρει υπηρεσίες και σε εξωτερικούς χώρους
- Υποστήριξη περισσότερων από μια κάμερα και διαχείριση τους σε πολλαπλά μόνιτορ
- Δυνατότητα όχι μόνο λήψης αλλά και έγγραφης σε αρχείο
- Ενημέρωση για νέο συμβάν με αποστολή email, ακόμη και με αποστολή sms στο κινητό του χρήστη
- Διαχείριση και φωτισμού εντός και εκτός του χώρου φύλαξης.

Γενικά είναι ένα σύστημα που θα παρέχει συνεχή επιτήρηση και ασφάλεια σε όλους τους χώρους .στους οποίους θα εφαρμοστεί .

---

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

---

- <http://www.netbeans.org>
- <http://java.sun.com/>
- <http://java.sun.com/docs/books/tutorial/>
- <http://java.sun.com/javase/technologies/desktop/media/jmf/>
- To appear in Smart Environments: Technologies, Protocols, and Applications ed. D.J. Cook and S.K. Das, John Wiley, New York, 2004
- Wireless Sensor Networks1. F. L. LEWIS
- JDDAC: <https://jddac.dev.java.net/>
- Getting Started with JDDAC – JddacClient , Glenn Engel, Jerry Liu
- IEEE 1451:[http://www.completetest.com/IEEE1451\\_overview.htm](http://www.completetest.com/IEEE1451_overview.htm)
- TEDS: <http://www.ni.com/teds/>
- 
- SORCER <http://sorcer.cs.ttu.edu/>
- Using Java Interaction Proxies for Web-based Steering of Distributed Simulations
- JavaReflectionAPI,  
<http://www.sun.com/products/jdk1.2/docs/guide/reflection>.
- JAVATM ON SENSORS  
John S. Nolan, Cristina Cifuentes  
Sun Labs
- Open Source Java<sup>a</sup> Media APIs
- Introduction to wireless Sensor and Control , Tom Bender , Tendril Networks
- Highlevel Realtime Programming in Java  
David F. Bacon Perry Cheng David Grove Michael Hind V.T. Rajan Eran Yahav, IBM T.J. Watson Research Center
- Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization  
Pratik Biswas, Yinyu Ye



- <http://www.securitylabs.gr/forum/showthread.php?p=19635>
- <http://dev.mysql.com/doc/refman/5.0/en/index.html>
- E-Mailing Through Java , Vipin Singla



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ



004000097397

ΠΤ - ΜΗΥΤ  
2008  
ΑΝΑ