



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Ανάπτυξη πληροφοριακού συστήματος για την διαχείριση
των ιατρικών αποβλήτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχαήλ Ε. Μυλωνάκης

Επιβλέπων: Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2013



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανάπτυξη πληροφοριακού συστήματος για την διαχείριση των ιατρικών αποβλήτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μιχαήλ Ε. Μυλωνάκης

Επιβλέπων: Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ... Ιουνίου 2013.

.....
Δ. Κουτσούρης
Καθηγητής Ε.Μ.Π.

.....
Δ. Φωτιάδης
Καθηγητής Ε.Μ.Π.

.....
Γ. Ματσόπουλος
Επ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2013

.....
Μιχαήλ Μ. Μυλωνάκης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός Ε.Μ.Π.

Copyright © 2013 Μιχαήλ Ε Μυλωνάκης.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στην παρούσα διπλωματική εργασία παρουσιάζεται το πληροφοριακό σύστημα “Medical Recycling” το οποίο πραγματεύεται το πολύ σημαντικό ζήτημα της διαχείρισης νοσοκομειακών αποβλήτων. Το συγκεκριμένο πληροφοριακό σύστημα αξιοποιεί τα δεδομένα που συγκεντρώνονται από τις υπάρχουσες δικτυακές υποδομές μιας αντιπροσωπευτικής νοσοκομειακής μονάδας αναφοράς καθώς επίσης και την δυνατότητα ταυτοποίησης μέσω ραδιοσυχνότητας (RFID tagging) των βασικών «οντοτήτων» του συστήματος, που εμπλέκονται στη διαχείριση των αποβλήτων, με απώτερο στόχο την ορθολογική οργάνωση και απλοποίηση των αντίστοιχων διαδικασιών.

Το σύστημα αναπτύχθηκε σε λειτουργικό σύστημα Windows 7 με το Visual Studio 2012, με χρήση της γλώσσας προγραμματισμού C# και χρησιμοποιώντας το framework ASP.NET MVC.

Λέξεις Κλειδιά

Πληροφοριακό σύστημα, Διαχείριση ιατρικών αποβλήτων, Ιστοσελίδα, Λογισμικό, ASP.NET, ASP.NET MVC, C#, RFID.

Abstract

In the following thesis information system “Medical Recycling” is presented. Medical Recycling deals with the very important issue of clinical waste management. This information system makes use of the existing data and networking facilities of a typical hospital unit, as well as the ability to authenticate basic system entities that are involved into waste management via Radiofrequency waves (RFID tagging), so that clinical waste procedures can become simplified and rational organized.

This information system was developed under the operating system Windows 7 using Visual Studio 2012, with the programming language C# and ASP.NET MVC framework.

Keywords

Information System, Clinical Waste Management, Web Application, Software, ASP.NET, ASP.NET MVC, C#, RFID.

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστώ τον κ. Κουτσούρη Δημήτριο για την δυνατότητα που μου προσέφερε να εκπονήσω την παρούσα διπλωματική εργασία. Επίσης ευχαριστώ θερμά τον κ. Αθανάσιο Αναστασίου για την άψογη συνεργασία και τις σημαντικές συμβουλές και παρατηρήσεις του που υπήρξαν καθοριστικές για την δημιουργία αλλά και την γενικότερη εικόνα του πληροφοριακού συστήματος και κατ επέκταση την συγγραφή της εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την υπομονή και την στήριξή τους.

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ	11
2 ΔΙΑΧΕΙΡΙΣΗ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ.....	13
2.1 ΚΑΤΗΓΟΡΙΕΣ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ	13
2.1.1 <i>Ιατρικά Απόβλητα Αστικού Χαρακτήρα</i>	<i>14</i>
2.1.2 <i>Επικίνδυνα Ιατρικά Απόβλητα αμιγώς Μολυσματικού Χαρακτήρα.....</i>	<i>14</i>
2.1.3 <i>Επικίνδυνα Ιατρικά Απόβλητα Μολυσματικού και Τοξικού Χαρακτήρα.....</i>	<i>14</i>
2.1.4 <i>Επικίνδυνα Ιατρικά Απόβλητα αμιγώς Τοξικού Χαρακτήρα.....</i>	<i>15</i>
2.1.5 <i>Άλλα Ιατρικά Απόβλητα.....</i>	<i>15</i>
2.2 ΔΙΑΔΙΚΑΣΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ	15
2.2.1 <i>Διαχωρισμός και Συλλογή Ιατρικών Αποβλήτων.....</i>	<i>16</i>
2.2.2 <i>Ενδονοσοκομειακή μεταφορά.....</i>	<i>19</i>
2.2.3 <i>Προσωρινή Αποθήκευση</i>	<i>19</i>
2.2.4 <i>Εξωνοσοκομειακή Μεταφορά</i>	<i>20</i>
3 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	21
3.1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	21
3.1.1 <i>Ανάλυση απαιτήσεων υποσυστήματος καταχώρησης των δεδομένων.....</i>	<i>21</i>
3.1.1.1 <i>Καταχώρηση ομάδων από περιέκτες και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί.....</i>	<i>22</i>
3.1.1.2 <i>Καταχώρηση νέων RFID readers, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί.....</i>	<i>23</i>
3.1.1.4 <i>Καταχώρηση νέων χώρων προσωρινής αποθήκευσης, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί.....</i>	<i>24</i>
3.1.1.5 <i>Εισαγωγή και διαγραφή οχήματος και οδηγού, καθώς επίσης και των εκτελούμενων δρομολογίων, και επεξεργασία των πληροφοριών που τους αφορούν....</i>	<i>24</i>
3.1.2 <i>Ανάλυση απαιτήσεων υποσυστήματος παρακολούθησης και διαχείρισης των ιατρικών αποβλήτων.....</i>	<i>24</i>
3.1.2.1 <i>Παρακολούθηση και παρουσίαση της κατάστασης των συλλεκτών, των απορριμμάτων, των αναγνωστών (readers), των χώρων προσωρινής αποθήκευσης και των οχημάτων</i>	<i>25</i>
3.1.2.2 <i>Έλεγχος των βασικών μεταβλητών του συστήματος και ειδοποίηση όταν παραβιάζεται κάποιος κανόνας ή οι τιμές των μεταβλητών πλησιάζουν τις οριακές</i>	<i>25</i>
3.2 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ - ΣΥΣΧΕΤΙΣΕΩΝ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	26
3.2.1 <i>Περιγραφή Οντοτήτων.....</i>	<i>26</i>
3.2.2 <i>Περιγραφή Συσχετίσεων.....</i>	<i>27</i>
4 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΤΟΥ Π.Σ	30
4.1 .NET FRAMEWORK	30
4.2 ASP.NET MVC.....	35
4.3 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C#.....	36

4.4 MASTER PAGES	37
4.5 AJAX ΚΑΙ JQUERY	38
4.6 CSS.....	39
4.7 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	39
4.7.1 Εργαλεία ανάπτυξης	39
4.7.2 Σχεσιακό Διάγραμμα της Βάσης Δεδομένων του Πληροφοριακού Συστήματος.....	40
4.7.2.1 Table dbo.Bag	42
4.7.2.2 Table dbo.Category	43
4.7.2.3 Table dbo.Bucket	43
4.7.2.4 Table dbo.Fridge	44
4.7.2.5 Table dbo.ClinicalWaste	45
4.7.2.6 Table dbo.Reader	46
4.7.2.7 Table dbo.Truck	47
4.7.2.8 Table BagBucket.....	48
4.7.2.9 Table BucketFridge.....	48
4.7.2.10 Table dbo.BucketReader	49
4.7.2.11 Table dbo.FridgeTruck	49
4.7.2.12 Table dbo.Route	50
4.7.2.13 Table dbo.Notifications.....	51
5 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ	52
5.1 ΣΥΝΔΕΣΗ (LOG IN) ΧΡΗΣΤΗ	57
5.2 ΕΓΓΡΑΦΗ (REGISTRATION) ΧΡΗΣΤΗ.....	62
5.3 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΠΕΡΙΕΚΤΩΝ (CONTAINERS)	68
5.3.1 Επισκόπηση, τροποποίηση και διαγραφή Περιεκτών.....	68
5.3.2 Καταχώρηση νέων περιεκτών.....	71
5.4 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΚΑΤΗΓΟΡΙΩΝ (CATEGORIES).....	74
5.4.1 Επισκόπηση και διαχείριση των κατηγοριών ιατρικών αποβλήτων.....	74
5.4.2 Καταχώρηση νέας κατηγορίας Ιατρικών Αποβλήτων	76
5.5 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΑΝΑΓΝΩΣΤΩΝ (READERS).....	76
5.5.1 Επισκόπηση, τροποποίηση και διαγραφή Αναγνώστων	77
5.5.2 Καταχώρηση νέου Αναγνώστη.....	77
5.5.3 Εμφάνιση των καταγραφών (recordings) του Αναγνώστη.....	78
5.5.4 Καταχώρηση νέας καταγραφής ενός Αναγνώστη	81
5.6 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΣΥΛΛΕΚΤΩΝ (BUCKETS)	82
5.6.1 Επισκόπηση, τροποποίηση και διαγραφή Συλλεκτών.....	83
5.6.2 Καταχώρηση νέου Συλλέκτη	83
5.6.3 Εισαγωγή ενός Περιέκτη σε έναν Συλλέκτη	85
5.6.4 Εκκένωση ενός Συλλέκτη σε έναν Χώρο Προσωρινής Αποθήκευσης (Storage Area) ..	88
5.7 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΧΩΡΩΝ ΠΡΟΣΩΡΙΝΗΣ ΑΠΟΘΗΚΕΥΣΗΣ (STORAGE AREAS)	91
5.7.1 Επισκόπηση, τροποποίηση και διαγραφή Χώρων Προσωρινής Αποθήκευσης.....	91
5.7.2 Καταχώρηση νέου χώρου προσωρινής αποθήκευσης	93
5.7.3 Εκκένωση ενός χώρου προσωρινής αποθήκευσης σε κάποιο όχημα μεταφοράς	94
5.8 ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΟΧΗΜΑΤΩΝ ΜΕΤΑΦΟΡΑΣ (VEHICLES) ΚΑΙ ΤΩΝ ΟΔΗΓΩΝ (DRIVERS)	96
5.8.1 Επισκόπηση, τροποποίηση και διαγραφή οχημάτων μεταφοράς και οδηγών.....	96
5.8.2 Καταχώρηση νέου οχήματος μεταφοράς και οδηγού	98
5.8.3 Διαχείριση δρομολογίων.....	99

5.9 ΕΙΔΟΠΟΙΗΣΕΙΣ (NOTIFICATIONS)	103
6 ΕΠΙΛΟΓΟΣ	107
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ	107
6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	107
7 ΒΙΒΛΙΟΓΡΑΦΙΑ	109
8 ΠΑΡΑΡΤΗΜΑ	112

ΚΕΦΑΛΑΙΟ

1

1 ΕΙΣΑΓΩΓΗ

Αναμφίβολα, τα κυριότερα χαρακτηριστικά γνωρίσματα της δεκαετίας που διανύουμε είναι η έκρηξη της πληροφορίας και της επικοινωνίας. Οι υποδομές δικτύωσης που παρέχονται είναι πλέον υψηλής ποιότητας και αξιοπιστίας, με αποτέλεσμα όλο και περισσότεροι φορείς και επιχειρήσεις να βασίζονται στα πληροφοριακά συστήματα για την οργάνωσή τους και τις υπηρεσίες που παρέχουν.

Η άνθιση αυτή των πληροφοριακών συστημάτων δεν θα μπορούσε να αφήσει ανεπηρέαστο τον κλάδο της υγείας, αφού αναμφισβήτητα αποτελεί ζήτημα που βρίσκεται στο επίκεντρο της προσοχής όλων των ανεπτυγμένων χωρών του κόσμου. Σημαντικό μέρος του προϋπολογισμού κάθε χώρας δαπανάται στον τομέα της υγείας, και κατ'επέκταση στην δημιουργία και συντήρηση πληροφοριακών συστημάτων που δραστηριοποιούνται στον χώρο αυτό.

Σκοπός μας στην συγκεκριμένη διπλωματική εργασία είναι η δημιουργία ενός πληροφοριακού συστήματος που θα βοηθήσει και θα απλοποιήσει την διαχείριση των ιατρικών αποβλήτων εντός και εκτός της νοσοκομειακής μονάδας. Παρουσιάζεται ο τρόπος με τον οποίο ο διαχειριστής του συστήματος μπορεί χρησιμοποιώντας το να διαχειρίζεται αποτελεσματικά τα ιατρικά απόβλητα που παράγονται στην μονάδα, και να παρακολουθεί ανά πάσα στιγμή την πορεία και την κατάσταση των απορριμμάτων εντός του νοσοκομείου.

Συγκεκριμένα στο κεφάλαιο 2 της παρούσας εργασίας, παρουσιάζονται οι διαδικασίες για την διαχείριση των αποβλήτων εντός της νοσοκομειακής μονάδας, οι περιορισμοί και οι υποχρεώσεις που επιβάλλονται λόγω της επικινδυνότητας πολλών εξ αυτών, και οι απαραίτητες υποδομές που πρέπει να υπάρχουν.

Στο κεφάλαιο 3 αναλύονται οι απαιτήσεις και οι προδιαγραφές του πληροφοριακού συστήματος, όπως αυτές εξάχθηκαν από την παρουσίαση των διαδικασιών στην πρώτη ενότητα, και βάσει αυτών γίνεται ο σχεδιασμός και η ανάλυσή του. Παρουσιάζονται και αναλύονται οι κύριες οντότητες του πληροφοριακού συστήματος, οι αλληλεπιδράσεις μεταξύ αυτών, και παραθέτονται τα αντίστοιχα διαγράμματα.

Στο κεφάλαιο 4 παρουσιάζεται η διαδικασία ανάπτυξης του πληροφοριακού συστήματος, το περιβάλλον ανάπτυξης, οι απαιτήσεις σε υλικό και λογισμικό και τα εργαλεία που χρησιμοποιήθηκαν.

Στο κεφάλαιο 5 παρουσιάζεται η λειτουργία του πληροφοριακού συστήματος μέσα από παραδείγματα χρήσης. Παρουσιάζονται οι αντίστοιχες εικόνες που δείχνουν πως μπορεί αυτός που διαχειρίζεται το πληροφοριακό σύστημα να το χρησιμοποιήσει αποτελεσματικά και να επωφεληθεί τα μέγιστα από αυτό.

Τέλος, στο παράρτημα παρουσιάζονται τα κυριότερα σημεία του κώδικα της εφαρμογής.

2 ΔΙΑΧΕΙΡΙΣΗ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ

2.1 ΚΑΤΗΓΟΡΙΕΣ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ

Με τον όρο ιατρικά απόβλητα νοούνται όλα εκείνα τα απόβλητα που παράγονται από νοσοκομειακές μονάδες, όπως για παράδειγμα δημόσια και ιδιωτικά θεραπευτήρια, στρατιωτικά νοσοκομεία, κέντρα υγείας, διαγνωστικά κέντρα κλπ. Το μεγαλύτερο ποσοστό εξ αυτών (περίπου 80%) θεωρούνται απόβλητα αστικού χαρακτήρα και είναι παρόμοια με τα οικιακά απορρίμματα και συνεπώς δεν αποτελούν κίνδυνο, και η διαχείρισή τους, δεδομένου ότι υπάρχει ανάγκη για διατήρηση του κόστους σε χαμηλά επίπεδα, πρέπει να είναι ίδια με αυτή των αστικών αποβλήτων. Το υπόλοιπο ποσοστό των αποβλήτων (περίπου 20%) θεωρείται επικίνδυνο για την υγεία, και άρα υπάρχει ανάγκη για ξεχωριστή διαχείρισή του [1].

Τα Ιατρικά Απόβλητα (Ι.Α) περιλαμβάνουν τις παρακάτω κατηγορίες [2]:

- **Ιατρικά Απόβλητα Αστικού Χαρακτήρα (ΙΑ-ΑΧ).**
- **Επικίνδυνα Ιατρικά Απόβλητα Αμιγώς Μολυσματικού Χαρακτήρα (ΕΙΑ-ΜΧ).**
- **Επικίνδυνα Ιατρικά Απόβλητα Μολυσματικού και Τοξικού Χαρακτήρα (ΕΙΑ-ΜΤΧ).**
- **Επικίνδυνα Ιατρικά Απόβλητα Αμιγώς Τοξικού Χαρακτήρα (ΕΙΑ-ΤΧ).**
- **Άλλα Ιατρικά απόβλητα.**

2.1.1 Ιατρικά Απόβλητα Αστικού Χαρακτήρα

Προέρχονται κυρίως από τις διοικητικές και τις οικιακού τύπου εργασίες (καθαριότητα, Παρασκευή φαγητού, εκτυπώσεις, κλπ) καθώς και από εργασίες συντήρησης των εγκαταστάσεων. Επίσης σε αυτήν την κατηγορία περιλαμβάνονται και τα Επικίνδυνα Ιατρικά Απόβλητα αμιγώς Μολυσματικού Χαρακτήρα που έχουν υποστεί επιτυχώς την διαδικασία αποστείρωσης. Ως παραδείγματα ιατρικών αποβλήτων αστικού χαρακτήρα, αναφέρονται ενδεικτικά απόβλητα από την Παρασκευή φαγητών, απόβλητα ιματισμού, χαρτί, γυαλί, χαρτόνι, πλαστικό, υλικά συσκευασίας κλπ.

2.1.2 Επικίνδυνα Ιατρικά Απόβλητα αμιγώς Μολυσματικού Χαρακτήρα

Προέρχονται από παθολογικά απόβλητα, όπως για παράδειγμα αίμα, ανθρώπινα υγρά, ανθρώπινοι ιστοί, ανθρώπινα μέλη, όργανα, έμβρυα κλπ. Ακόμη θεωρούνται τα αιχμηρά ιατρικά εργαλεία όπως βελόνες, νυστέρια, λάμες, σπασμένα γυαλιά κλπ. Τα εργαλεία αυτά πρέπει να θεωρούνται μολυσματικά απόβλητα, είτε είναι μολυσμένα είτε όχι, και είναι ιδιαίτερα επικίνδυνα αφού προκαλούν τα περισσότερα ατυχήματα. Τα απόβλητα αυτά περιέχουν πιθανώς παθογόνους οργανισμούς (ιούς, μύκητες, βακτήρια κλπ) σε συγκεντρώσεις ή ποσότητες ικανές να προκαλέσουν ασθένειες.

2.1.3 Επικίνδυνα Ιατρικά Απόβλητα Μολυσματικού και Τοξικού Χαρακτήρα

Περιλαμβάνουν απόβλητα από ανάπτυξη ερευνητικών δραστηριοτήτων τα οποία περιέχουν εκτός από κάποιον μολυσματικό παράγοντα και κάποια τοξική χημική ουσία, όπως αντιδραστήριο διαλύτη κλπ. Παραδείγματα τέτοιου είδους αποβλήτων είναι πλάκες, τρυβλία

καλλιέργειας, χημειοθεραπείες, κυτταροστατικά φάρμακα, χρησιμοποιημένους ορούς με κυτταροστατικά φάρμακα κλπ.

2.1.4 Επικίνδυνα Ιατρικά Απόβλητα αμιγώς Τοξικού Χαρακτήρα

Περιλαμβάνουν απόβλητα που περιέχουν επικίνδυνες κυτταροτοξικές ή κυτταροστατικές ή άλλες φαρμακευτικές ουσίες. Παραδείγματα τέτοιου είδους αποβλήτων είναι φαρμακευτικά απόβλητα όπως ληγμένα φάρμακα, μη χρησιμοποιημένα ή μολυσμένα φαρμακευτικά προϊόντα, καθώς επίσης και τα κουτιά ή τα μπουκάλια που τα περιείχαν, απόβλητα που περιέχουν υδράργυρο, αμίαντο, χρησιμοποιημένα έλαια, απολυμαντικά, οξέα, διαλυτικά κλπ.

2.1.5 Άλλα Ιατρικά Απόβλητα

Άλλα ιατρικά απόβλητα θεωρούνται τα απόβλητα που δεν περιλαμβάνονται στις προηγούμενες κατηγορίες, όπως για παράδειγμα υγρά απόβλητα που προσομοιάζονται με αστικά λύματα, μπαταρίες, δοχεία υπό πίεση, ραδιενεργά απόβλητα κλπ.

2.2 ΔΙΑΔΙΚΑΣΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΙΑΤΡΙΚΩΝ ΑΠΟΒΛΗΤΩΝ

Η διαχείριση των Ιατρικών αποβλήτων έχει ως στόχο να τα καταστήσει αβλαβή για την Δημόσια Υγεία και το περιβάλλον και περιλαμβάνει τις ακόλουθες φάσεις – στάδια [3]:

- Διαχωρισμός και Συλλογή Ιατρικών Αποβλήτων
- Ενδονοσοκομειακή μεταφορά
- Προσωρινή αποθήκευση
- Εξωνοσοκομειακή μεταφορά

2.2.1 Διαχωρισμός και Συλλογή Ιατρικών Αποβλήτων

Διαχωρισμός και συλλογή είναι η συγκέντρωση και η τοποθέτηση των ιατρικών αποβλήτων σε κατάλληλη συσκευασία, που ονομάζεται περιέκτης, ανάλογα με τον τύπο των ιατρικών αποβλήτων. Η διαδικασία αυτή πραγματοποιείται στους χώρους που παράγονται τα απόβλητα.

Στην φάση του διαχωρισμού, τα απόβλητα διαχωρίζονται και τοποθετούνται στους κατάλληλους περιέκτες. Οι περιέκτες μπορεί να είναι είτε κοινές πλαστικές σακούλες (για τα αστικού τύπου απόβλητα), είτε ενισχυμένες σακούλες κατάλληλου πάχους και συγκεκριμένου χρώματος (συνήθως χρησιμοποιούνται κίτρινο και κόκκινο χρώμα), είτε ειδικά κουτιά. Να σημειωθεί πως η διαδικασία διαχωρισμού είναι πολύ σημαντική τόσο για την διασφάλιση της δημόσιας υγείας με την έννοια ότι στα επικίνδυνα απόβλητα επιβάλλεται ειδική μεταχείριση, όσο και για την διατήρηση του κόστους στα αναμενόμενα επίπεδα, αφού τα ακίνδυνα απόβλητα πρέπει να μεταχειρίζονται ως τέτοια και όχι σαν επικίνδυνα, γιατί κάτι τέτοιο θα ανέβαζε σημαντικά το κόστος.

Ο κάθε περιέκτης έχει μια ταυτότητα ραδιοσυχνότητας (RFID tag), για να είναι δυνατή η καταγραφή του εντός της νοσοκομειακής μονάδας.

Στις εικόνες που ακολουθούν παρουσιάζονται ενδεικτικά μερικοί περιέκτες.



Εικόνα 2.1. Κόκκινος Περιέκτης.



Εικόνα 2.2. Κίτρινος Περιέκτης.



Εικόνα 2.3. Περιέκτης – κουτί (hospital box).

Στην φάση της συλλογής, τα απόβλητα τοποθετούνται ανάλογα με τον τύπο τους στους κατάλληλους περιέκτες, και οι περιέκτες με την σειρά τους τοποθετούνται σε συλλέκτες, που μπορεί να είναι είτε τυπικοί κάδοι απορριμμάτων για τα απόβλητα αστικού χαρακτήρα, είτε συγκεκριμένα καλάθια, είτε ειδικά τροχήλατα καρότσια.

Οι συλλέκτες έχουν ενσωματωμένους αναγνώστες ραδιοσυχνότητας (RFID readers) έτσι ώστε να καταγράφεται ποιοι περιέκτες εισήλθαν σε αυτούς. Ακόμη, καταγράφεται η ακριβής

ώρα και ημερομηνία που εισήλθε ο κάθε περιέκτης στον συλλέκτη, το βάρος του, και η θερμοκρασία στο εσωτερικό του.

Προφανώς, οι περιέκτες δεν μπορούν να μείνουν στους συλλέκτες για μεγάλο χρονικό διάστημα, αλλά πρέπει να εκκενώνονται ανά τακτά χρονικά διαστήματα, και αυτό εξαρτάται από την κατηγορία των απορριμμάτων που περιέχουν.

Στις εικόνες που ακολουθούν παρουσιάζονται ενδεικτικά μερικοί συλλέκτες.



Εικόνα 2.4. Συλλέκτης Ιατρικών Αποβλήτων.



Εικόνα 2.5. Τροχήλατο καρότσι Ιατρικών αποβλήτων



Εικόνα 2.6. Τυπικός κάδος απορριμμάτων.

2.2.2 Ενδονοσοκομειακή μεταφορά

Ενδονοσοκομειακή μεταφορά είναι οι εργασίες για την κατάλληλη περισυλλογή και ασφαλή μεταφορά των απορριμμάτων από τον χώρο παραγωγής τους στον χώρο προσωρινής αποθήκευσης. Οι χώροι προσωρινής αποθήκευσης μπορεί να είναι είτε ειδικά διαμορφωμένα δωμάτια, είτε μεγάλα ντουλάπια, είτε ψυγεία, ανάλογα με τα απόβλητα θα περιέχουν. Η μεταφορά των απορριμμάτων σε αυτά γίνεται μεταφέροντας τους αντίστοιχους συλλέκτες που τα περιέχουν. Γενικά δεν επιτρέπεται η μεταφορά των περιεκτών με τα χέρια. Σε διάφορα προκαθορισμένα σημεία ενδιαφέροντος του νοσοκομείου είναι τοποθετημένοι οι αναγνώστες ραδιοσυχνότητας (RFID readers) έτσι ώστε να παρακολουθούνται οι κινήσεις των συλλεκτών. Οι συλλέκτες έχουν για αυτό τον σκοπό, δικό τους RFID tag έτσι ώστε να είναι δυνατή η καταγραφή τους από τους αναγνώστες ανά πάσα χρονική στιγμή.

2.2.3 Προσωρινή Αποθήκευση

Ο τελικός προορισμός των συλλεκτών, όπως αναφέρθηκε και προηγουμένως είναι οι χώροι προσωρινής αποθήκευσης. Οι χώροι αυτοί έχουν ενσωματωμένους αναγνώστες έτσι ώστε να

μπορεί να παρακολουθείται ποιοι συλλέκτες αδειάζονται σε κάθε τέτοιο χώρο, και η αντίστοιχη ώρα και ημερομηνία που συμβαίνει. Ακόμη, έχουν την δυνατότητα να λαμβάνουν μετρήσεις για την θερμοκρασία στο εσωτερικό τους, και το βάρος των απορριμμάτων που περιέχουν. Να σημειωθεί σε αυτό το σημείο ότι η θερμοκρασία ενός ψυγείου ή γενικότερα αποθηκευτικού χώρου αυξάνεται ανάλογα με τον όγκο των απορριμμάτων που βρίσκονται στο εσωτερικό του. Για την διασφάλιση της σωστής λειτουργίας ψύξης – κλιματισμού, θεωρούμε ότι η επιθυμητή θερμοκρασία είναι κάτω από ένα κατώφλι, και όταν η θερμοκρασία ανέβει πάνω από αυτό, το σύστημα ψύξης – κλιματισμού ενεργοποιείται έτσι ώστε να επανέλθει στα επιθυμητά επίπεδα.

2.2.4 Εξωνοσοκομειακή Μεταφορά

Εξωνοσοκομειακή μεταφορά είναι η μεταφορά από τον χώρο προσωρινής αποθήκευσης της νοσοκομειακής μονάδας, προς τον χώρο τελικής διάθεσης. Για την διαδικασία της μεταφοράς προσλαμβάνονται ειδικά οχήματα, τα οποία μπορούν να διατηρούν την θερμοκρασία του χώρου όπου φορτώνονται τα απορρίμματα στις επιτρεπτές τιμές, και έχουν δυνατότητα συνεχής καταγραφής της θερμοκρασίας και του βάρους των απορριμμάτων.

3 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

3.1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η επιτυχία οποιουδήποτε έργου λογισμικού εξαρτάται σε μεγάλο βαθμό από τον αρχικό εντοπισμό των απαιτήσεων του υπό ανάπτυξη συστήματος τόσο μέσα από την λήψη των ζητούμενων, όσο και από την ανάδειξη των διαφόρων υπο-προβλημάτων που ενδέχεται να προκύψουν. Το πρώτο βήμα στην ανάπτυξη ενός έργου λογισμικού είναι η ανάλυση απαιτήσεων (requirement analysis). Τα αποτελέσματα του βήματος αυτού είναι οι τεχνολογικές, επιχειρηματικές και λειτουργικές προδιαγραφές [4].

Σε αυτό το κεφάλαιο παρουσιάζεται η βασική λειτουργικότητα και οι σχεδιαστικές αποφάσεις που αφορούν την υλοποίηση του υποσυστήματος παρακολούθησης και διαχείρισης των ιατρικών αποβλήτων, καθώς και του υποσυστήματος καταχώρησης δεδομένων που είναι απαραίτητα για την λειτουργία του πρώτου. Πιο συγκεκριμένα παρουσιάζονται οι απαιτήσεις που πρέπει να ικανοποιεί το πληροφοριακό σύστημα, και οι βασικές λειτουργίες και οι κανόνες που επιβάλλει η λειτουργία του για τα δύο αυτά υποσυστήματα.

3.1.1 Ανάλυση απαιτήσεων υποσυστήματος καταχώρησης των δεδομένων

Το υποσύστημα καταχώρησης των δεδομένων υλοποιεί την εισαγωγή των δεδομένων του συστήματός μας στην βάση δεδομένων, με σκοπό την μετέπειτα επεξεργασία και παρουσίασή

τους από το υποσύστημα παρακολούθησης και διαχείρισης. Βασική απαίτηση από το υποσύστημα αυτό είναι η σωστή καταχώρηση και επεξεργασία των πληροφοριών. Από τις πληροφορίες αυτές πρέπει να προκύπτει ο τύπος των ιατρικών αποβλήτων που είναι σχεδιασμένος να περιέχει ο κάθε περιέκτης, το χρώμα του περιέκτη, το είδος του (πλαστική σακούλα, κουτί κλπ), το RFID tag του, η ημερομηνία παραλαβής και λήξης του και το μέγεθός του. Ακόμη, πρέπει να προκύπτει η θέση, των RFID readers, των χώρων προσωρινής αποθήκευσης, και των συλλεκτών, τα διαθέσιμα φορτηγά στα οποία θα εκκενώνονται οι χώροι προσωρινής αποθήκευσης της νοσοκομειακής μονάδας, καθώς επίσης και οι οδηγοί αυτών.

Σύμφωνα με τις απαιτήσεις που περιγράφηκαν στην προηγούμενη παράγραφο εξάγεται η απαιτούμενη λειτουργικότητα που πρέπει να παρέχει το υποσύστημα :

- Καταχώρηση ομάδων από περιέκτες, μαζί με τις απαραίτητες πληροφορίες που σχετίζονται με αυτά, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί.
- Καταχώρηση νέων αναγνωστών (RFID readers) και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων για αυτούς που έχουν ήδη εισαχθεί.
- Προσθήκη νέων συλλεκτών και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και διόρθωση προηγούμενων λανθασμένων στοιχείων.
- Προσθήκη νέων χώρων προσωρινής αποθήκευσης, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί.
- Εισαγωγή και διαγραφή οχήματος (φορτηγού) και οδηγού οχήματος, και επεξεργασία πληροφοριών που τον αφορούν.

3.1.1.1 Καταχώρηση ομάδων από περιέκτες και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί

Στο σενάριο αυτό, ο χρήστης που είναι υπεύθυνος για την παραλαβή και την καταχώρηση των περιεκτών (θα μπορούσε να είναι ο υπεύθυνος προμηθειών – αποθήκης της νοσοκομειακής μονάδας ή και ο διαχειριστής του συστήματος), επιχειρεί να καταχωρήσει την ημερομηνία παραλαβής τους, τον τύπο τους, το χρώμα τους, την ημερομηνία λήξης τους, την κατηγορία ιατρικών αποβλήτων που μπορούν να εισαχθούν σε αυτά, το μέγεθός τους, καθώς επίσης και να τους αναθέσει, κατά ομάδες ή σε κάθε ένα ξεχωριστά, RFID tags. Προφανώς υπάρχει πρόβλεψη ώστε να μπορεί να διορθώσει κάποιο λανθασμένο στοιχείο που εισήγαγε. Ακόμη, θα μπορεί να προσθέσει μία νέα κατηγορία ιατρικών αποβλήτων ή να επεξεργαστεί κάποια ήδη υπάρχουσα.

Οι ενέργειες αυτές θα μπορούσαν να πραγματοποιηθούν είτε από τον σταθμό εργασίας του υπεύθυνου προμηθειών , είτε από κάποιο φορητό μέσο (πχ laptop , pda) , που θα είναι συνδεδεμένο στο δίκτυο.

3.1.1.2 Καταχώρηση νέων RFID readers, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί

Στο σενάριο αυτό , ο διαχειριστής του συστήματος καταχωρεί τους RFID readers στην βάση δεδομένων , περιγράφοντας την τοποθεσία τους και όποια άλλη πληροφορία θεωρεί χρήσιμη.

3.1.1.3 Καταχώρηση νέων συλλεκτών, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί

Στο σενάριο αυτό , κατ' αναλογία με το ακριβώς προηγούμενο , ο χρήστης καταχωρεί τους συλλέκτες (που , προφανώς , έχουν ενσωματωμένους readers) συνοδευόμενους από την τοποθεσία τους και την χωρητικότητά τους . Να σημειωθεί σε αυτό το σημείο ότι οι συλλέκτες θεωρείται ότι έχουν αισθητήρες για να μετράνε το βάρος και την θερμοκρασία σε τακτά χρονικά διαστήματα , καθώς είναι κομβικής σημασίας πληροφορίες για την σωστή διαχείριση των ιατρικών αποβλήτων από το σύστημά μας, σύμφωνα με όσα περιγράφηκαν στο προηγούμενο κεφάλαιο. Θεωρείται ακόμα ότι υπάρχει η απαραίτητη δικτυακή υποδομή έτσι ώστε αυτές οι μετρήσεις να στέλνονται στην βάση δεδομένων , και να ανανεώνονται τα αντίστοιχα πεδία στους πίνακες της βάσης δυναμικά.

3.1.1.4 Καταχώρηση νέων χώρων προσωρινής αποθήκευσης, και όλων των απαραίτητων πληροφοριών που σχετίζονται με αυτούς, και επεξεργασία πιθανών πιθανών λανθασμένων στοιχείων που έχουν ήδη εισαχθεί

Στο σενάριο αυτό, ο χρήστης καταχωρεί τους χώρους προσωρινής αποθήκευσης της νοσοκομειακής μονάδας, περιγράφοντας την τοποθεσία τους και την χωρητικότητά τους. Οι χώροι αυτοί περιέχουν αισθητήρες έτσι ώστε να λαμβάνουν μετρήσεις για την θερμοκρασία και το βάρος των απορριμμάτων που περιέχουν όπως περιγράφηκε στο προηγούμενο κεφάλαιο, και οι μετρήσεις αυτές στέλνονται στην βάση δεδομένων.

3.1.1.5 Εισαγωγή και διαγραφή οχήματος και οδηγού, καθώς επίσης και των εκτελούμενων δρομολογίων, και επεξεργασία των πληροφοριών που τους αφορούν

Στο σενάριο αυτό, ο χρήστης καταχωρεί, τα στοιχεία για τα φορτηγά και τους οδηγούς, με δυνατότητα να διαγράψει παλιές καταχωρήσεις. Ακόμη καταχωρεί τα δρομολόγια των οχημάτων κατά την διαδικασία μεταφοράς των απορριμμάτων από τους χώρους προσωρινής αποθήκευσης της νοσοκομειακής μονάδας, στον χώρο τελικής διαχείρισής τους. Να σημειωθεί σε αυτό το σημείο ότι, εκτός από τις πληροφορίες που εισάγονται για το κάθε όχημα (φορτηγό), όπως το μοντέλο του, η χωρητικότητά του και η όποια περιγραφή για αυτό, θεωρείται ότι έχουν αισθητήρες και μετράνε την θερμοκρασία, το καθαρό βάρος των αποβλήτων, καθώς επίσης και δέκτη GPS, και υπάρχει η απαραίτητη δικτυακή υποδομή έτσι ώστε οι μετρήσεις αυτές να στέλνονται ανά τακτά χρονικά διαστήματα στο σύστημά μας, και να ανανεώνονται δυναμικά τα αντίστοιχα πεδία στον πίνακα της βάσης.

3.1.2 Ανάλυση απαιτήσεων υποσυστήματος παρακολούθησης και διαχείρισης των ιατρικών αποβλήτων.

Το υποσύστημα παρακολούθησης και διαχείρισης ιατρικών αποβλήτων παίρνει σαν είσοδο τα δεδομένα που προκύπτουν από το υποσύστημα καταχώρησης, και τα παρουσιάζει ομαδοποιημένα στους ενδιαφερόμενους χρήστες, έτσι ώστε να συναχθούν συμπεράσματα για την κατάσταση των συλλεκτών, την θέση των φορτηγών, την θερμοκρασία στο περιβάλλον που βρίσκονται οι περιέκτες, τις καταχωρήσεις των Readers και γενικά όποιο άλλο χαρακτηριστικό των οντοτήτων του πρώτου υποσυστήματος είναι άξια προσοχής. Ακόμη πρέπει να ενημερώνει τους υπεύθυνους, όταν η θερμοκρασία σε κάποιο χώρο προσωρινής αποθήκευσης πλησιάζει την

μέγιστη θερμοκρασία στην οποία επιτρέπεται να εκτεθούν τα απορρίμματα , ή όταν πλησιάζει το χρονικό όριο στο οποίο μπορούν να μείνουν εκτός των χώρων προσωρινής αποθήκευσης. Επίσης πρέπει να παρέχεται ειδοποίηση για πιθανούς περιέκτες οι οποίοι έχουν λήξει.

Σύμφωνα με τις απαιτήσεις που περιγράφηκαν στην προηγούμενη παράγραφο εξάγεται η απαιτούμενη λειτουργικότητα που πρέπει να παρέχει το υποσύστημα :

- Παρακολούθηση της κατάστασης των συλλεκτών με τους περιέκτες, των προσωρινών αποθηκευτικών χώρων, των καταχωρήσεων των readers , των φορτηγών (και την κατάσταση στον χώρο του φορτηγού που βρίσκονται τα απόβλητα – πχ θερμοκρασία, βάρος), και παρουσίαση των παραπάνω πληροφοριών σε όλους τους ενδιαφερόμενους χρήστες (που , προφανώς , έχουν ταυτοποιηθεί, και έχουν το δικαίωμα να τις βλέπουν).
- Ενημέρωση – ειδοποίηση όταν παραβιάζεται κάποιος κανόνας από αυτούς που επιβάλλει το business logic του πληροφοριακού συστήματος που αναπτύσσουμε.

3.1.2.1 Παρακολούθηση και παρουσίαση της κατάστασης των συλλεκτών, των απορριμμάτων, των αναγνώστων (readers), των χώρων προσωρινής αποθήκευσης και των οχημάτων

Στο σενάριο αυτό, ο χρήστης χρησιμοποιεί το πληροφοριακό σύστημα και βλέπει σημαντικές πληροφορίες για την κατάσταση του συστήματος , όπως την πληρότητα των συλλεκτών , την θερμοκρασία στο εσωτερικό τους, την διαδρομή τους, την κατάσταση των χώρων προσωρινής αποθήκευσης και γενικά των περιεκτών εντός της νοσοκομειακής μονάδας, και την κατάσταση των οχημάτων μεταφοράς εκτός αυτής.

3.1.2.2 Έλεγχος των βασικών μεταβλητών του συστήματος και ειδοποίηση όταν παραβιάζεται κάποιος κανόνας ή οι τιμές των μεταβλητών πλησιάζουν τις οριακές

Στο σενάριο αυτό ο alarm agent – που δεν είναι φυσικό πρόσωπο , αλλά ένας δράστης που ζει στο περιβάλλον του υποσυστήματος παρακολούθησης και διαχείρισης , παρακολουθεί την κατάσταση των κρίσιμων μεταβλητών του συστήματος , όπως για παράδειγμα την θερμοκρασία στους χώρους προσωρινής αποθήκευσης , τον χρόνο στον οποίο τα απόβλητα βρίσκονται στους συλλέκτες, πριν μεταφερθούν στους χώρους προσωρινής αποθήκευσης, την ημερομηνία λήξης των περιεκτών, και εμφανίζει τις αντίστοιχες ειδοποιήσεις στον χρήστη.

3.2 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ - ΣΥΣΧΕΤΙΣΕΩΝ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στην ενότητα αυτή δίνεται η περιγραφή του μοντέλου Οντοτήτων – Συσχετίσεων (Entity-Relationship Diagram) από το οποίο υλοποιείται το Σχεσιακό Διάγραμμα της Βάσης Δεδομένων του Πληροφοριακού Συστήματος.

3.2.1 Περιγραφή Οντοτήτων

Οι οντότητες του Πληροφοριακού Συστήματος είναι οι παρακάτω:

- *Περιέκτης (table Bag):* Αναπαριστά τους περιέκτες που καταχωρούνται στην βάση δεδομένων . Οι ιδιότητες που περιέχει αντιστοιχούν σε όλες τις απαραίτητες πληροφορίες για τον κάθε περιέκτη, που είναι το είδος του, το χρώμα του, το RFID tag του , το μέγεθός του , η ημερομηνία παραλαβής του, η ημερομηνία λήξης του και η κατάστασή του (αν είναι άδειο ή όχι).
- *Κατηγορία Ιατρικών αποβλήτων (table Category):* Αναπαριστά τις κατηγορίες ιατρικών αποβλήτων που διαχειρίζεται το νοσοκομείο , με σκοπό να υπάρχει γνώση για το τι περιέχει (και τι θα μπορεί να περιέχει) ένας περιέκτης. Οι ιδιότητες που περιέχει αντιστοιχούν στο όνομα της κατηγορίας , και οποιαδήποτε άλλη πληροφορία θεωρείται χρήσιμη για αυτήν.
- *Συλλέκτες (table bucket):* Αναπαριστά όλους τους διαθέσιμους συλλέκτες αποβλήτων της νοσοκομειακής μονάδας. Οι ιδιότητες που περιέχει αντιστοιχούν στις απαραίτητες για την λειτουργία του συστήματος πληροφορίες , που είναι το καθαρό βάρος του συλλέκτη , η θερμοκρασία του , που ανανεώνονται δυναμικά στην βάση , από τις μετρήσεις που στέλνουν οι αισθητήρες, η χωρητικότητά του , η τοποθεσία στην οποία βρίσκονται, ένα όνομα για τον συλλέκτη, και το RFID tag του.
- *Χώροι Προσωρινής Αποθήκευσης (table Fridge):* Αναπαριστά όλους τους διαθέσιμους χώρους προσωρινής αποθήκευσης της νοσοκομειακής μονάδας. Οι ιδιότητες που περιέχει αντιστοιχούν σε ένα σύντομο όνομα για αυτούς, την χωρητικότητά τους, την τοποθεσία τους, το βάρος τους και την θερμοκρασία τους, που ανανεώνονται δυναμικά στην βάση και τους συλλέκτες οι οποίοι έχουν εκκενωθεί σε αυτούς.
- *Αναγνώστης RFID (table Reader):* Αναπαριστά όλους τους αναγνώστες (readers) που βρίσκονται εντός και εκτός του νοσοκομείου . Οι ιδιότητες που περιέχει είναι η τοποθεσία στην οποία βρίσκονται, και ένα όνομα που τον περιγράφει.
- *Ιατρικά απόβλητα (table Clinical Waste):* Αναπαριστά όλα τα απορρίμματα που δημιουργήθηκαν στο περιβάλλον του νοσοκομείου , και είτε έχουν μεταφερθεί στα φορτηγά για διαχείριση , είτε είναι ακόμα μέσα σε κάποιον συλλέκτη (ή χώρο προσωρινής

αποθήκευσης) εντός νοσοκομείου. Παρέχει δηλαδή κάποιας μορφής ιστορικό για το σύστημά μας. Οι ιδιότητες που περιέχει αντιστοιχούν στο RFID tag από τον περιέκτη που το φιλοξενούσε, την ημερομηνία δημιουργίας του, την ημερομηνία καταστροφής του (αν έχει πάει για διαχείριση), το συλλέκτη στον οποίο βρίσκεται (ή, αν έχει καταστραφεί, το ID του τελευταίου συλλέκτη στον οποίο βρισκόταν), το δρομολόγιο (όχημα, οδηγός, και ημερομηνία) κατά το οποίο απομακρύνθηκε από τον χώρο προσωρινής αποθήκευσης της νοσοκομειακής μονάδας, αν κάτι τέτοιο έχει συμβεί.

- *Όχημα (table Truck)*: Αναπαριστά όλα τα διαθέσιμα οχήματα που συνεργάζονται με το νοσοκομείο, και έρχονται σε αυτό για να παραλάβουν τα ιατρικά απόβλητα και να τα διαχειριστούν. Οι ιδιότητες που περιέχει αντιστοιχούν στο όνομα (μοντέλο) του οχήματος, την χωρητικότητά του, το καθαρό βάρος, την θερμοκρασία, και την τρέχουσα θέση (που προκύπτουν από τους αισθητήρες, GPS που έχει το φορητό, και αποστέλλονται δυναμικά στην βάση δεδομένων), και οποιαδήποτε άλλη πληροφορία για αυτό θεωρείται χρήσιμη.
- *Οδηγός (table Driver)*: Αναπαριστά όλους τους διαθέσιμους οδηγούς που συνεργάζονται με το νοσοκομείο, και είναι οι οδηγοί των φορητών στις διάφορες διαδρομές. Οι ιδιότητες που περιέχει αντιστοιχούν, στις απαραίτητες πληροφορίες για τον οδηγό, δηλαδή το ονοματεπώνυμό του, την διεύθυνσή του, τα τηλέφωνα του, το e-mail του, και γενικά οποιαδήποτε άλλη παρατήρηση θεωρείται χρήσιμη.

3.2.2 Περιγραφή Συσχετίσεων

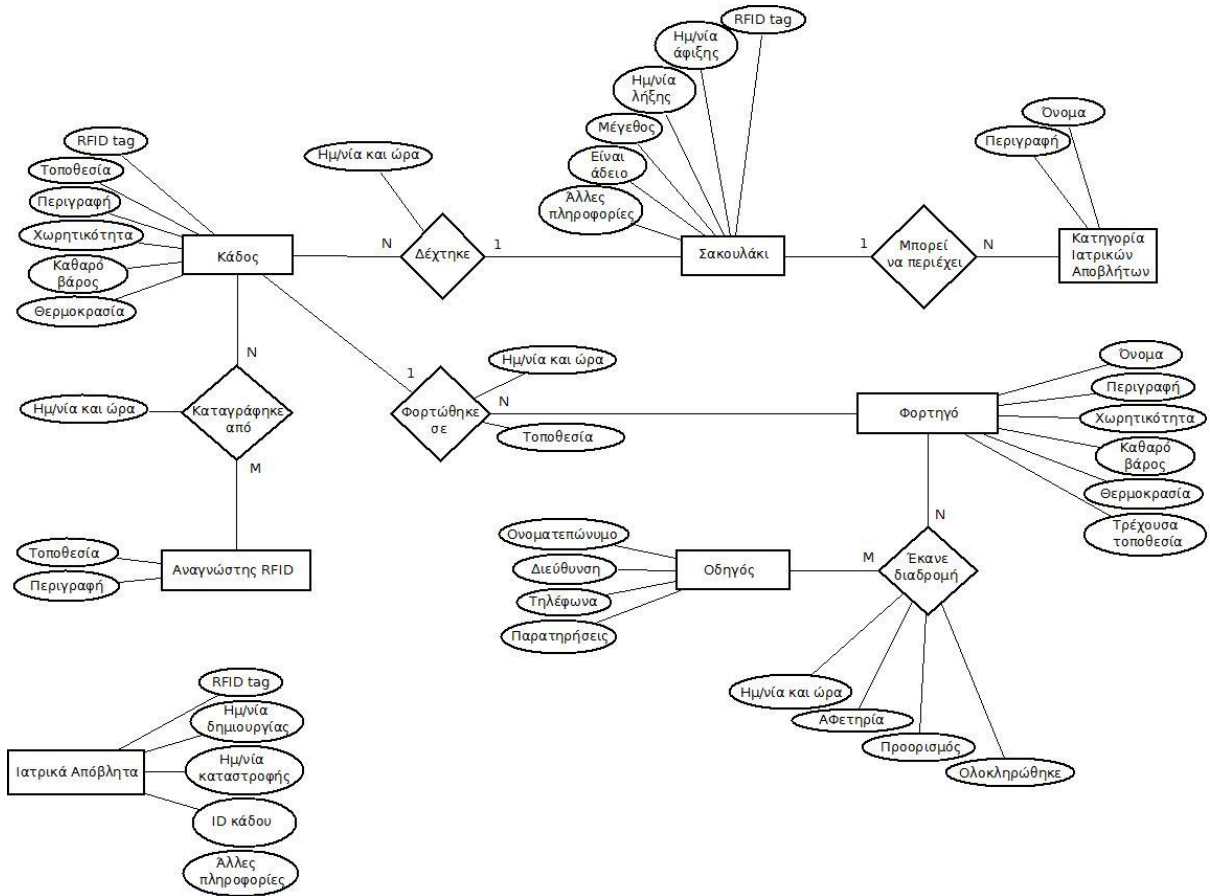
Οι βασικές συσχετίσεις του Πληροφοριακού Συστήματος είναι οι παρακάτω:

- *Μπορεί να περιέχει*: Αναπαριστά τις κατηγορίες ιατρικών αποβλήτων που μπορεί να περιέχει ο κάθε περιέκτης. Η συμμετοχή της οντότητας περιέκτης στην σχέση είναι υποχρεωτική, καθώς κάθε περιέκτης πρέπει να ανήκει σε μία κατηγορία ιατρικών αποβλήτων, αλλά από την μεριά της κατηγορίας ιατρικών αποβλήτων είναι προαιρετική, καθώς υπάρχει περίπτωση σε κάποια χρονική περίοδο, να μην υπάρχουν περιέκτες κάποιας συγκεκριμένης κατηγορίας ιατρικών αποβλήτων. Η σχέση είναι 1:N αφού μία κατηγορία μπορεί να αντιστοιχεί σε πολλούς περιέκτες, αλλά κάθε περιέκτης μπορεί να περιέχει 1 τύπο ιατρικών αποβλήτων.
- *Δέχτηκε*: Αναπαριστά την σχέση που έχει ο περιέκτης με τον συλλέκτη, που πρακτικά είναι όταν εισέρχεται ένας περιέκτης σε αυτόν. Η σχέση περιέχει την ιδιότητα ημερομηνία και ώρα, που αντιστοιχεί στην ώρα που εισήλθε ο περιέκτης στον συλλέκτη. Η συμμετοχή και των 2 οντοτήτων είναι προαιρετική, καθώς δεν είναι απαραίτητο όλοι οι περιέκτες να βρίσκονται σε κάποιον συλλέκτη (μπορεί να μην έχουν χρησιμοποιηθεί ακόμα), και δεν είναι ανάγκη όλοι οι συλλέκτες να έχουν κάποιον περιέκτη στο εσωτερικό τους (μπορεί

κάποιος να είναι άδειος). Η σχέση είναι 1:N αφού ένας περιέκτης μπορεί να είναι μέσα σε ένα μόνο συλλέκτη, αλλά ένας περιέκτης μπορεί να περιέχει παραπάνω του ενός περιέκτες.

- *Εκκενώθηκε σε:* Αναπαριστά την σχέση που έχει ο συλλέκτης με τον χώρο προσωρινής αποθήκευσης, δηλαδή την διαδικασία εκκένωσης κάποιου συλλέκτη σε αυτόν. Η σχέση περιέχει την ιδιότητα ημερομηνία και ώρα, που αντιστοιχεί στην χρονική στιγμή κατά την οποία εκκενώθηκε ο συλλέκτης στον χώρο προσωρινής αποθήκευσης. Η συμμετοχή και των 2 οντοτήτων είναι προαιρετική, καθώς δεν είναι απαραίτητο όλοι οι συλλέκτες να έχουν εκκενωθεί σε κάποιο χώρο προσωρινής αποθήκευσης, και δεν είναι ανάγκη όλοι οι χώροι προσωρινής αποθήκευσης να περιέχουν απορρίμματα από κάποιον συλλέκτη. Η σχέση είναι 1:N αφού ένας συλλέκτης μπορεί να εκκενωθεί σε ένα μόνο χώρο προσωρινής αποθήκευσης, αλλά ένας χώρος προσωρινής αποθήκευσης μπορεί να περιέχει απορρίμματα από παραπάνω του ενός συλλέκτες.
- *Καταγράφηκε από:* Αναπαριστά την σχέση που έχει ένας συλλέκτης με τον αναγνώστη (reader) που καταγράφει την έλευσή του από το σημείο που βρίσκεται. Η σχέση περιέχει την ιδιότητα ημερομηνία και ώρα, που αντιστοιχεί την ώρα που καταγράφηκε ο συλλέκτης από τον αναγνώστη. Η συμμετοχή και των 2 σχέσεων είναι προαιρετική, αφού δεν είναι απαραίτητο ένας αναγνώστης να έχει καταγράψει την κίνηση κάποιου συλλέκτη και δεν είναι ανάγκη κάποιος κάδος να έχει καταγραφεί από κάποιον reader. Η σχέση είναι N:M αφού, γενικά, πολλοί συλλέκτες μπορούν να περάσουν από πολλούς διαφορετικούς readers.
- *Φορτώθηκε σε:* Αναπαριστά την σχέση μεταξύ του χώρου προσωρινής αποθήκευσης και του οχήματος στο οποίο αυτός φορτώνεται. Η σχέση περιέχει την ιδιότητα ημερομηνία και ώρα, που αντιστοιχεί την ώρα που φορτώθηκε ο χώρος προσωρινής αποθήκευσης στο όχημα μεταφοράς, και η τοποθεσία στην οποία έγινε η φόρτωση. Η συμμετοχή και των 2 σχέσεων είναι προαιρετική, αφού δεν είναι απαραίτητο ένας χώρος προσωρινής αποθήκευσης να έχει φορτωθεί σε κάποιο όχημα, και δεν είναι ανάγκη σε ένα όχημα να έχει φορτωθεί κάποιος χώρος προσωρινής αποθήκευσης. Η σχέση είναι 1:N αφού ένας κάδος φορτώνεται σε 1 μόνο όχημα, αλλά σε ένα όχημα μπορούν να έχουν φορτωθεί πολλοί κάδοι.
- *Έκανε διαδρομή:* Αναπαριστά την σχέση μεταξύ του οχήματος μεταφοράς και του οδηγού που το χειρίζεται, με σκοπό την διαχείριση των αποβλήτων. Η σχέση περιέχει την ιδιότητα ημερομηνία και ώρα, την τοποθεσία αφετηρίας, τον προορισμό, και την κατάσταση της διαδρομής (αν έχει ολοκληρωθεί ή όχι). Η συμμετοχή και των 2 σχέσεων είναι προαιρετική, αφού ένα όχημα ή ένας οδηγός μπορεί να μην έχει πραγματοποιήσει διαδρομή. Η σχέση είναι N:M αφού γενικά, σε διαφορετικά χρονικά διαστήματα προφανώς, μπορούν οι διάφοροι οδηγοί να οδηγούν διαφορετικά οχήματα.

Τα παραπάνω απεικονίζονται στο παρακάτω σχήμα του ER διαγράμματος του Πληροφοριακού Συστήματος.



Εικόνα 3.1. E-R Diagram.

4 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΤΟΥ Π.Σ

Το παρόν πληροφοριακό σύστημα αναπτύχθηκε σε λειτουργικό σύστημα Windows 7, με χρήση της γλώσσας προγραμματισμού C# που είναι μέρος του .NET Framework. Πιο συγκεκριμένα, χρησιμοποιήθηκε η τεχνολογία και τα εργαλεία που παρέχει το ASP.NET MVC Framework, μέσω του ολοκληρωμένου περιβάλλοντος ανάπτυξης Visual Studio 2012 for Web. Για την βάση δεδομένων του πληροφοριακού συστήματος χρησιμοποιήθηκε ο Microsoft SQL Server 2010.

4.1 .NET FRAMEWORK

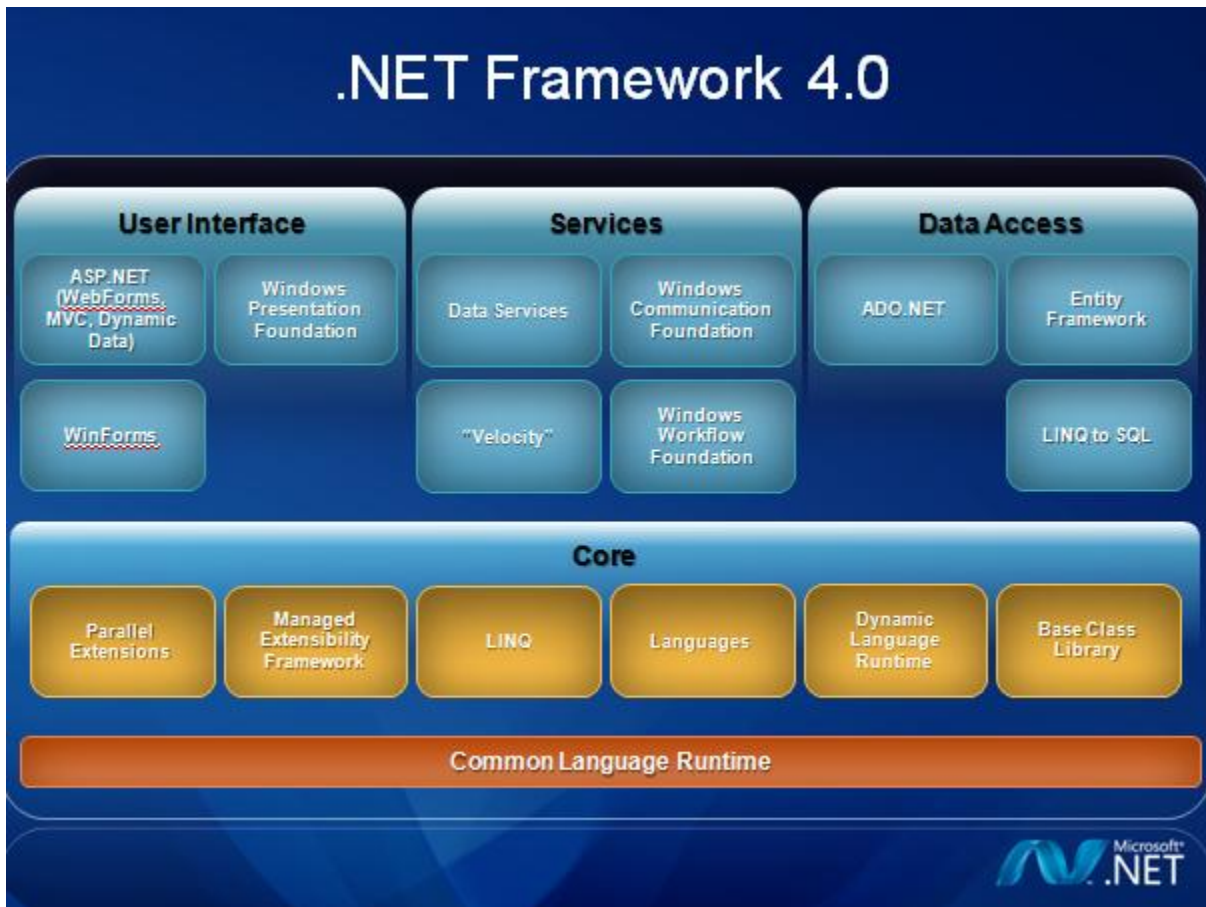
Το .NET Framework [5] είναι μια πλατφόρμα ανάπτυξης λογισμικού, για την δημιουργία εφαρμογών που προορίζονται για τα λειτουργικά συστήματα Windows, Windows phone, Windows Server και Windows Azure. Αποτελείται από την Common Language Runtime (CLR) , που ουσιαστικά είναι ένα εικονικό περιβάλλον εκτέλεσης των προγραμμάτων .NET, και την βιβλιοθήκη κλάσεων .NET Framework class library. Γενικά, το .NET Framework παρέχει ένα ευέλικτο αλλά ταυτόχρονα ελεγχόμενο περιβάλλον εκτέλεσης εφαρμογών, ευκολία στην ανάπτυξη εφαρμογών και project, και ταυτόχρονη χρήση πολλών γλωσσών προγραμματισμού, οι οποίες είναι οι .NET Languages.

Πιο συγκεκριμένα οι βασικές ψηφίδες του .NET Framework είναι οι ακόλουθες:

- .NET Framework Base Class Library. Περιλαμβάνει πολλές βιβλιοθήκες κλάσεων με σκοπό να παρέχουν έτοιμο κώδικα στους προγραμματιστές, για λειτουργίες που θεωρείται ότι είναι συχνά χρησιμοποιούμενες, και μπορούν να μοντελοποιηθούν

αποτελεσματικά. Παραδείγματα τέτοιων λειτουργιών είναι η εγγραφή και ανάγνωση σε αρχεία ή στην οθόνη, η επικοινωνία με βάσεις δεδομένων, η δημιουργία γραφικών στοιχείων κλπ.

- .NET Languages. Είναι ένα σύνολο γλωσσών προγραμματισμού που μπορούν να συνδυαστούν, και στο ίδιο Project, και η επικοινωνία μεταξύ τους. Αυτές οι γλώσσες είναι οι C#, Visual Basic, JS script, F#, J#, C++. Γενικά οι γλώσσες αυτές ακολουθούν τις αρχές του αντικειμενοστραφούς προγραμματισμού.
- Language Integrated Query (LINQ). Είναι μία ψηφίδα του .NET Framework που παρέχει την δυνατότητα διεξαγωγής ερωτημάτων σε δεδομένα (querying capabilities) στις γλώσσες του .NET. Αυτή η δυνατότητα είναι πολύ σημαντική, αφού τα περισσότερα πληροφοριακά συστήματα είναι στενά συνδεδεμένα με δεδομένα σε κάποια βάση δεδομένων, και το γεγονός ότι παρέχεται φυσική υποστήριξη από την γλώσσα (δυνατότητα αυτόματης συμπλήρωσης, επισήμανση λαθών, σύνθετες πράξεις με τα δεδομένα) αυξάνει σημαντικά την παραγωγικότητα των προγραμματιστών.
- ASP.NET. Το ASP.NET είναι ένα server-side Web application Framework που έχει σχεδιαστεί με σκοπό να παρέχει την ικανότητα στους Web Developers να δημιουργούν δυναμικές ιστοσελίδες. Είναι ένα σχετικά καινούριο Framework, που εμφανίστηκε το 2002. Είναι δομημένο στο Common Language Runtime (CLR), και επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν την .NET γλώσσα προγραμματισμού της αρεσκείας τους. Το ASP.NET υποστηρίζει τρία διαφορετικά μοντέλα ανάπτυξης, που είναι τα Web Pages, Web Forms, και το MVC που είναι και αυτό που χρησιμοποιήθηκε για την ανάπτυξη του παρόντος πληροφοριακού συστήματος.
- Intermediate Language (IL). Μία μορφή byte code, που αποτελεί ενδιάμεση μορφή μεταγλώττισης, σε ένα εικονικό περιβάλλον εκτέλεσης, γνωστό ως CLR, και σκοπό έχει ο κώδικας που παράγεται να είναι ανεξάρτητος της αρχιτεκτονικής του εκάστοτε επεξεργαστή.
- Common Language Runtime (CLR). Το εικονικό περιβάλλον (virtual machine), που ελέγχει και εκτελεί τις .NET εφαρμογές.
- Garbage Collection. Το εικονικό περιβάλλον CLR είναι υπεύθυνο για την αποδέσμευση των μη χρησιμοποιούμενων πόρων που δεσμεύουν οι .NET εφαρμογές.
- Common Type System (CTS), και Common Language Specification (CLS). Είναι ένα σύνολο από κανόνες οι οποίοι έχουν σκοπό να καταστήσουν δυνατή την επικοινωνία μεταξύ των διαφορετικών .NET Languages. Με αυτόν τον τρόπο είναι δυνατή ή εύκολη μετατροπή κώδικα από μια γλώσσα σε μία άλλη, αφού ουσιαστικά οι τύποι των δεδομένων (τα αντικείμενα) υπακούουν σε κοινούς κανόνες, και έχουν το ίδιο interface προς τον προγραμματιστή.



Εικόνα 4.1. Αρχιτεκτονική του .NET Framework

Πριν επεκταθούμε σε μια πιο αναλυτική παρουσίαση της δομής της πλατφόρμας .NET και των τεχνολογιών που την περιβάλλουν και χρησιμοποιήθηκαν στην ανάπτυξη του παρόντος πληροφοριακού συστήματος είναι σκόπιμο να παρουσιάσουμε τα σημαντικότερα χαρακτηριστικά και πλεονεκτήματά της:

- *Υποστήριξη πολλαπλών γλωσσών.* Στο .NET, η διαλειτουργικότητα μεταξύ των γλωσσών επιτυγχάνεται μέσω της διαγλωσσικής κληρονομικότητας. Μαζί με ένα ενοποιημένο σύστημα τύπων, η συνένωση κώδικα γραμμένου σε διαφορετικές γλώσσες είναι πλέον εύκολη υπόθεση. Αυτό διευκολύνει και άλλα προγραμματιστικά πρότυπα όπως το συναρτησιακό της F#.
- *Ανάπτυξη βασισμένη σε components.* Η δημιουργία ανεξάρτητων κομματιών λογισμικού που περιέχουν βιβλιοθήκες κλάσεων και διαμοιράσιμα τμήματα λειτουργικότητας μιας εφαρμογής έχει γίνει πολύ πιο εύκολη από ότι ήταν παλιότερα. Η διαμοιράσιμη μονάδα

κώδικα στο .NET βασίζεται στην έννοια του assembly, το οποίο μεταφέρει πληροφορίες που απαιτούνται για τη χρήση της όπως η έκδοση της.

- *Μεταφερσιμότητα.* Η Μεταφερσιμότητα (portability) επιτυγχάνεται μέσω της εικονικής μηχανής στο CLR. Η εικονική μηχανή αυτή είναι η προδιαγραφή μιας αφηρημένης εικονικής μηχανής για την οποία κώδικας γραμμένος σε οποιαδήποτε από τις .NET γλώσσες μεταγλωττίζεται σε μια ενδιάμεση γλώσσα γνωστή ως Common Intermediate Language (CIL) και περιέχεται σε ένα assembly. Ένα assembly είναι είτε ένα εκτελέσιμο αρχείο (exe) είτε ένα dll. Η μεταγλώττιση σε αυτήν την ενδιάμεση γλώσσα είναι που εξασφαλίζει και την ανεξαρτησία από την πλατφόρμα, η οποία ωστόσο μένει μόνο στη θεωρία, αφού η μοναδική πλήρης υλοποίηση του .NET είναι διαθέσιμη μόνο για την πλατφόρμα των Windows. Όμως είναι διαθέσιμη μια μερική υλοποίηση ανοιχτού κώδικα με το όνομα Mono.
- *Απλή εγκατάσταση εφαρμογών.* Αντίθετα με τα components που βασίζονται στο COM, δεν χρειάζεται καμία καταχώρηση στο registry του συστήματος για την εγκατάσταση των assemblies καθώς μια απλή αντιγραφή αρκεί για τη χρήση τους. Επιπρόσθετα τα πιθανά προβλήματα της χρήσης των Dynamic Link Libraries (DLLs) απαλείφθηκαν με την υποστήριξη της ύπαρξης πολλαπλών εκδόσεων του ίδιου component. Το ίδιο το .NET Framework είναι ένα καλό παράδειγμα αυτής της δυνατότητας, αφού είναι δυνατόν να υπάρχουν πολλαπλές εκδόσεις του framework εγκατεστημένες στο ίδιο μηχανήμα.
- *Επικοινωνία με υπάρχων κώδικα.* Παρόλο που η πλατφόρμα .NET προορίζεται να είναι ένα υποκατάστατο των παλαιότερων τεχνολογιών, υποστηρίζει τη χρήση κομματιών λογισμικού που έχουν αναπτυχθεί σε παλιότερες τεχνολογίες όπως της COM και παρέχει πρόσβαση σε λειτουργίες χαμηλού επιπέδου του λειτουργικού συστήματος μέσω του μηχανισμού P/Invoke.
- *Αξιοπιστία.* Το .NET είναι σχεδιασμένο για τη δημιουργία λογισμικού υψηλής αξιοπιστίας (robustness). Πραγματοποιεί εκτεταμένους ελέγχους κατά τη μεταγλώττιση αλλά και κατά την εκτέλεση των προγραμμάτων. Τα χαρακτηριστικά του .NET από μόνα τους ωθούν τους προγραμματιστές στην απόκτηση αξιόπιστων προγραμματιστικών συνηθειών. Το μοντέλο διαχείρισης της μνήμης είναι απλό: η δέσμευση μνήμης γίνεται με μια λέξη (new), δεν υπάρχουν δείκτες προς θέσεις μνήμης σαν τύποι δεδομένων, ούτε αριθμητική δεικτών ενώ η αποδέσμευση μνήμης γίνεται αυτόματα μέσω του μηχανισμού garbage collection. Αυτό το απλό μοντέλο διαχείρισης μνήμης απαλείφει ολόκληρες κατηγορίες λαθών που απασχολούν τους προγραμματιστές σε C και C++. Η ανάπτυξη προγραμμάτων γίνεται με την πεποίθηση ότι αρκετά λάθη θα εντοπιστούν από το σύστημα πολύ πριν την ολοκλήρωσή της.
- *Αυτόματη διαχείριση μνήμης.* Άλλο ένα σημαντικό χαρακτηριστικό του .NET Framework είναι η δυνατότητα αυτόματης διαχείρισης της μνήμης μέσω του Garbage Collector. Ιστορικά η διαχείριση της μνήμης ήταν από τις δυσκολότερες εργασίες που είχε να κάνει ένας προγραμματιστής. Οι παλαιότερες προσεγγίσεις στη διαχείριση μνήμης ήταν είτε χαμηλού επιπέδου (όπως η malloc/free στη C και η new/delete στη C++), προκαλώντας διάφορα bugs, είτε αρκετά πολύπλοκη (όπως στην περίπτωση του COM). Ένας από τους στόχους του .NET και του CLR πιο συγκεκριμένα ήταν να απαλείψει από τους προγραμματιστές την ανάγκη διαχείρισης της μνήμης. Το CLR με το μηχανισμό Garbage Collector διαχειρίζεται τη μνήμη ερευνώντας πότε μπορεί να ελευθερωθεί με ασφάλεια μνήμη που δεν χρειάζεται άλλο σε ένα πρόγραμμα. Η μνήμη δεσμεύεται με την αρχικοποίηση των διαφόρων τύπων (αντικειμένων) από ένα διαθέσιμο τμήμα της μνήμης

που διαχειρίζεται το CLR γνωστό ως heap (σωρός). Όσο υπάρχει κάποια άμεση ή έμμεση (μέσω του γράφου των αντικειμένων) αναφορά προς ένα αντικείμενο, τότε αυτό θεωρείται πως χρησιμοποιείται. Στην αντίθετη περίπτωση, όταν δεν μπορεί να αναφερθεί από κάποιο άλλο τμήμα του κώδικα τότε είναι διαθέσιμο για καταστροφή. Ο garbage collector τρέχει ανά τακτά χρονικά διαστήματα σε ένα ξεχωριστό thread, εντοπίζει ποια αντικείμενα είναι έτοιμα προς καταστροφή και αποδεσμεύει τη μνήμη που αυτά χρησιμοποιούσαν.

- *Ασφάλεια.* Το .NET είναι σχεδιασμένο να λειτουργεί σε κατανεμημένα περιβάλλοντα, η ασφάλεια (security) των οποίων είναι υψίστης σημασίας. Με μέτρα που είναι ενσωματωμένα στις γλώσσες αλλά και κατά την εκτέλεση των προγραμμάτων, το .NET επιτρέπει τη δημιουργία προγραμμάτων που δεν μπορούν να προσβληθούν από έξω. Σε δικτυακά περιβάλλοντα οι εφαρμογές .NET είναι ασφαλείς από εισβολές μη εξουσιοδοτημένου κώδικα που επιχειρεί να δράσει στο παρασκήνιο και να εγκαταστήσει κακόβουλο λογισμικό ή να εισβάλει σε συστήματα αρχείων. Το περιβάλλον εκτέλεσης του .NET έχει ενσωματωμένο έναν μηχανισμό ασφάλειας με το όνομα Code Access Security (CAS) που απομονώνει τον κώδικα ανάλογα με την προέλευσή του, τον δημιουργό του ή άλλες πολιτικές. Το μοντέλο ασφάλειας συμβαδίζει με τους μηχανισμούς ασφάλειας που παρέχει το λειτουργικό σύστημα όπως οι Access Control Lists και τα Windows security tokens.
- *Απόδοση.* Η ενδιάμεση γλώσσα στην οποία μεταγλωττίζονται τα προγράμματα στο .NET, μεταγλωττίζεται πάντα με τη μέθοδο Just-in-Time (JIT). Ο μεταγλωττιστής JIT μεταγλωττίζει κάθε τμήμα του κώδικα στην ενδιάμεση γλώσσα όταν αυτό καλείται (just in time). Όταν έχει ήδη μεταγλωττιστεί κάποιο τμήμα κώδικα, το αποτέλεσμα αποθηκεύεται μέχρι η εφαρμογή να τερματίσει έτσι ώστε να μην χρειάζεται να ξαναμεταγλωττιστεί όταν ζητηθεί το συγκεκριμένο τμήμα. Η διαδικασία αυτή είναι αποδοτικότερη σε σύγκριση με την απευθείας μεταγλώττιση ολόκληρου του κώδικα, γιατί υπάρχει περίπτωση μεγάλα τμήματα κώδικα να μην εκτελεστούν ποτέ στην πραγματικότητα. Με τον μεταγλωττιστή JIT τέτοιος κώδικας δεν θα μεταγλωττίζονταν ποτέ.
- *Υποστήριξη με επαγγελματικά εργαλεία.* Το .NET Framework συνοδεύεται από ένα σύνολο εργαλείων για την υποστήριξη της ανάπτυξης λογισμικού σε αυτό, όπως το σύστημα σχεδιασμού, τα frameworks για unit testing, την πλατφόρμα για build MSBuild και το σύστημα debugging, μέσα από το Visual Studio 2010. Το Visual Studio είναι ένα από τα καλύτερα Integrated Development Environments (IDEs). Χρησιμοποιείται για την ανάπτυξη όλων των ειδών εφαρμογών που μπορούν να αναπτυχθούν στο .NET: από console εφαρμογές και εφαρμογές με γραφικές διεπαφές (GUIs) μέχρι web εφαρμογές και web services σε managed αλλά και εγγενή κώδικα. Οι Premium και Ultimate εκδόσεις του Visual Studio στοχεύουν στην ομαδική ανάπτυξη μαζί με τον Team Foundation Server (TFS) που χρησιμοποιείται για την παρακολούθηση ενός έργου .NET, τη διαχείριση του κώδικα, τη διαχείριση των αναφορών αλλά και το διαμοιρασμό κώδικα, τεκμηριώσεων ή άλλων εγγράφων.

4.2 ASP.NET MVC

Το MVC (Model – View – Controller) [6] είναι ένα μοντέλο αρχιτεκτονικής, που διαχωρίζει την εφαρμογή σε τρία κύρια συστατικά: το Model, το View, και τον Controller. Το ASP.NET MVC Framework ακολουθεί αυτήν την αρχιτεκτονική ανάπτυξης και αποτελεί ένα ελαφρύ και με μεγάλες δυνατότητες συντήρησης και ελεγχιμότητας Framework, που διατηρεί όλα τα πλεονεκτήματα του ASP.NET όπως τα λεγόμενα Master Pages ή το membership based authentication.

Ήδη, αρκετοί προγραμματιστές προτιμούν αυτήν την αρχιτεκτονική και έχουν υιοθετήσει το συγκεκριμένο Framework αφού η χρήση του ενδείκνυται για πλήθος εφαρμογών. Οι τρεις βασικές ψηφίδες, όπως αναφέρθηκαν και στην προηγούμενη παράγραφο, είναι οι ακόλουθες:

- **Model.** Είναι τα αντικείμενα εκείνα που υλοποιούν το business logic της εφαρμογής. Αποτελούν δηλαδή τις κύριες οντότητες της εφαρμογής, αφού η κατάσταση των αντικειμένων αυτών προσδιορίζει την συμπεριφορά της εφαρμογής. Τα αντικείμενα του μοντέλου συχνά αντιστοιχούν σε πίνακες κάποιας βάσης δεδομένων, οπότε ανακτούν τις τιμές από την βάση δεδομένων, ενεργούν πάνω στα δεδομένα, και στέλνουν πίσω τις ανανεωμένες πληροφορίες.
- **Views.** Είναι οι ψηφίδες που είναι υπεύθυνες για την παρουσίαση του User Interface (UI) της εφαρμογής. Τυπικά, αυτό το UI παράγεται από το μοντέλο.
- **Controllers.** Είναι οι ψηφίδες που είναι υπεύθυνες για την αλληλεπίδραση με τον χρήστη. Λαμβάνουν δηλαδή την είσοδο του χρήστη, επεξεργάζονται το μοντέλο (model) βάσει αυτής, και επιλέγουν ποιο view θα εμφανιστεί τελικά στον χρήστη. Ο controller δηλαδή χειρίζεται την είσοδο από τον χρήστη, την μεταφέρει στο μοντέλο, που με την σειρά του ενεργεί πάνω στα δεδομένα (στην βάση δεδομένων). Τέλος επιλέγεται ποιο View θα εμφανιστεί στον χρήστη.

Το MVC Framework όπως είδαμε κάνει σαφή διαχωρισμό των διαφορετικών τμημάτων κάθε εφαρμογής, δηλαδή των Input logic, business logic και UI Logic, και ταυτόχρονα προσφέρει έναν εύκολο και ευέλικτο τρόπο επικοινωνίας μεταξύ τους. Παρέχει δηλαδή ένα είδος καθοδήγησης στον προγραμματιστή σχετικά με το που πρέπει να βρίσκεται το κάθε τμήμα μέσα στην εφαρμογή, γεγονός που οδηγεί σε ευανάγνωστο και εύκολα συντηρήσιμο κώδικα, και ελαττώνει την πολυπλοκότητα αφού επιτρέπει στον προγραμματιστή να επικεντρώνεται σε ένα κομμάτι της εφαρμογής κάθε φορά (model ή view ή controller). Ακόμη, η προσέγγιση αυτή καθιστά εύκολη την ανάπτυξη μεγάλων project από ομάδες πολλών ατόμων. Γενικά, το ASP.NET MVC προσφέρει :

- Σαφή διαχωρισμό των διαδικασιών της εφαρμογής (Input Logic – Business Logic – UI Logic), Test Driven Development (TDD) και υψηλή συντηρησιμότητα της εφαρμογής.
- Ένα επεκτάσιμο και παραμετροποιήσιμο Framework. Όλες οι ψηφίδες του είναι σχεδιασμένες με τέτοιο τρόπο, ώστε να μπορούν εύκολα να αντικαθίστανται ή να τροποποιούνται. Μπορούμε για παράδειγμα να κατασκευάσουμε και να συνδέσουμε την δικιά μας View engine, Routing Policy και γενικά διάφορες ψηφίδες.
- Έναν ισχυρό μηχανισμό URL – mapping , που ξεφεύγει από την κλασσική προσέγγιση της αναζήτησης κάποιου αρχείου σε απομακρυσμένο server. Τα URL's δεν αντιπροσωπεύουν τα αρχεία με τις επεκτάσεις τους (πχ home.php) , αλλά είναι ευανάγνωστα και εύκολα ανακτήσιμα, και σχεδιασμένα να υποστηρίζουν διάφορα URL naming patterns που επωφελούνται από το λεγόμενο Search Engine Optimization (SEO) και το Representational State Transfer (REST) addressing.
- Υποστήριξη για την χρήση ισχυρής γλώσσας προγραμματισμού σε κάποια ιστοσελίδα, Master Pages (θα παρουσιαστούν αναλυτικά παρακάτω).

4.3 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C#

Η C# [7] είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, συνεπώς είναι υψηλού επιπέδου. Αυτό σημαίνει πως είναι φιλική προς τον προγραμματιστή, αφού αποκρύπτει τις λεπτομέρειες του υλικού του υπολογιστικού συστήματος, και παρέχει έναν ενιαίο τρόπο επικοινωνίας προς αυτό, μέσω των έτοιμων κλάσεων και βιβλιοθηκών που προσφέρει. Αποτελεί μια συνεχώς εξελισσόμενη γλώσσα, και σε κάθε νέα της έκδοση προστίθενται νέα χαρακτηριστικά και συντακτικό, με στόχο να κάνει τα πράγματα ευκολότερα για τον προγραμματιστή. Ακόμη, μέσω του ASP.NET μπορεί να ενσωματωθεί κώδικας σε C# ανάμεσα σε τυπικό κώδικα HTML με ξεκάθαρο και αποτελεσματικό τρόπο, βοηθώντας την κατασκευή δυναμικών ιστοσελίδων. Εν συντομία, τα βασικά χαρακτηριστικά της είναι:

- *Απλότητα (Simplicity).* Η C# είναι μια απλή γλώσσα που μπορεί να χρησιμοποιηθεί χωρίς εντατική εκμάθηση, ενώ ταυτόχρονα είναι εναρμονισμένη με σύγχρονες προγραμματιστικές πρακτικές. Οι θεμελιώδεις αρχές της γλώσσας μπορούν να κατανοηθούν γρήγορα κάτι που σημαίνει ότι οι προγραμματιστές θα είναι παραγωγικοί σε σύντομο χρονικό διάστημα. Η C# έχει σχεδιαστεί έτσι ώστε να μειώνεται η πιθανότητα πρόκλησης λαθών από την πολυπλοκότητα του κώδικα, αφού τη μειώνει σε μεγάλο βαθμό με το απλουστευμένο συντακτικό της και την οργάνωση κώδικά της.
- *Αντικειμενοστρέφεια (Object – Orientation).* Η C# από τα θεμέλια της σχεδιάστηκε να είναι αντικειμενοστραφής. Ο αντικειμενοστραφής προγραμματισμός επικράτησε σαν προγραμματιστικό πρότυπο την προηγούμενη δεκαετία και παραμένει στις πρώτες

προτιμήσεις των προγραμματιστών. Οι ανάγκες για καταναμημένα συστήματα πελάτη -εξυπηρετητή συμπίπτουν με την ενθουλάκωση και την ανταλλαγή μηνυμάτων που είναι βασικά χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού. Κατά πολλούς ειδικούς στις γλώσσες προγραμματισμού, η επιτυχής λειτουργία των προγραμματιστικών συστημάτων σε δικτυακά περιβάλλοντα αυξανόμενης πολυπλοκότητας βασίζεται στην Αντικειμενοστρέφεια. Η C# παρέχει μια ξεκάθαρη και αποδοτική αντικειμενοστραφή πλατφόρμα παρέχοντας στους προγραμματιστές μια συλλογή βιβλιοθηκών δοκιμασμένων αντικειμένων που παρέχουν λειτουργικότητα που ποικίλει από απλούς τύπους δεδομένων, σε διεπαφές εισόδου/εξόδου ή δικτυακές και εργαλεία για τη δημιουργία παραθυρικών εφαρμογών. Αυτές οι βιβλιοθήκες μπορούν να προσαρμοστούν στις ανάγκες του προγραμματιστή. Επιπρόσθετα η C# υποστηρίζει και τον προγραμματισμό βασισμένο σε components (component – based programming) ο οποίος επιτρέπει τον προσδιορισμό αυτόνομων μονάδων λειτουργικότητας (components) που είναι απομονωμένα και τεκμηριωμένα, παρουσιάζοντας ένα μοντέλο με ιδιότητες, μεθόδους, events και μεταδεδομένα για το component. Η C# υποστηρίζει αυτά τα χαρακτηριστικά άμεσα κάνοντας έτσι τη διαδικασία δημιουργίας και χρήσης των components πολύ εύκολη.

- *Οικειότητα (Familiarity)*. Στο ξεκίνημα της υλοποίησής της, οι δημιουργοί της τεχνολογίας C# απέρριψαν την ολοκληρωτική χρήση της C++ σαν γλώσσα υλοποίησης. Στη νέα γλώσσα ωστόσο, κράτησαν αρκετά χαρακτηριστικά της C++ αλλά και της Java και αφαίρεσαν την άχρηστη πολυπλοκότητα και των δύο. Έτσι έχοντας κρατήσει αρκετά από τα αντικειμενοστραφή χαρακτηριστικά και τη γενική φιλοσοφία της C++ αλλά και τη γενική ευκολία της Java βελτιώνοντας ορισμένα σημεία της, είναι σχετικά εύκολη τη «μετακόμιση» στη C# δεδομένου ότι η C η C++ αλλά και η Java είναι ευρέως γνωστές και χρησιμοποιούνται συχνά.

4.4 MASTER PAGES

Σχεδόν σε όλες τις επαγγελματικές web εφαρμογές, οι σελίδες που εμφανίζονται στον χρήστη έχουν την ίδια υφή, και μοιάζουν μεταξύ τους έτσι ώστε η πλοήγηση σε αυτές να είναι ευχάριστη και εύκολη. Μπορεί για παράδειγμα να υπάρχει μια μπάρα στο πάνω μέρος της σελίδας, ή ένα sidebar στα αριστερά και μέσα από αυτά να μπορεί ο χρήστης να πλοηγηθεί στην ιστοσελίδα. Προφανώς η συγγραφή ίδιου HTML κώδικα για όλες τις ιστοσελίδες (στην περίπτωση του MVC, Views) δεν έχει νόημα, και μειώνει την παραγωγικότητα του προγραμματιστή.

Διαφορετικά Frameworks προσφέρουν διαφορετικά εργαλεία για την αποφυγή του παραπάνω σεναρίου. Στην περίπτωση του ASP.NET MVC το εργαλείο αυτό ονομάζεται Master Pages. Τα Master Pages μπορούν να θεωρηθούν ως μέρη ιστοσελίδων, τα οποία καλούνται από όσα Views θέλουν να τα συμπεριλάβουν. Συνεπώς, εφ' όσον επιθυμούμε ομοιομορφία σε όλα τα Views της εφαρμογής, συμπεριλαμβάνουμε ένα Master Page σε όλα τα Views, και το κάθε View

αναφέρεται στο όνομα του Master Page που θέλει να συμπεριλάβει και παράγεται αυτόματα ο κώδικας HTML που χρειάζεται.

Στο παρών πληροφοριακό σύστημα χρησιμοποιήθηκε η τεχνολογία των Master Pages, για να επιτύχουμε την ομοιομορφία σε όλα τα Views, και παρέχεται ένα sidebar που επιτρέπει στον χρήστη την πλοήγηση στον ιστότοπο, ένα κύριο μενού σε μορφή μπάρας, links που τον παραπέμπουν στην σελίδα διαχείρισης του προφίλ του, και στην σελίδα αποσύνδεσης (log off) και background image με την εικόνα και το logo της εφαρμογής μας. Οι εικόνες που θα παρουσιαστούν στο 5^ο κεφάλαιο θα αποσαφηνίσουν αυτά που μόλις περιγράψαμε.

4.5 AJAX και jQuery

Με τον όρο AJAX [8] (Asynchronous JavaScript and XML) αναφερόμαστε σε μια τεχνολογία που επιτρέπει την δυναμική ανανέωση μέρους ή ολόκληρης της ιστοσελίδας, χωρίς να χρειάζεται να την ανανεώσει ο χρήστης, αλλά μέσω των λεγόμενων ασύγχρονων requests. Ο πελάτης (client) ανακτά τα δεδομένα από τον εξυπηρετητή (server) στο παρασκήνιο, χωρίς να είναι εμφανής στον χρήστη αυτή η λειτουργία. Η τεχνολογία αυτή συνετέλεσε τα μέγιστα στην ανάπτυξη δυναμικών, υψηλής ποιότητας ιστοσελίδων.

Επειδή συχνά παρεξηγείται ο όρος, να σημειωθεί πως το AJAX είναι απλά μια τεχνολογία, και όχι κάποια γλώσσα προγραμματισμού. Συγκεκριμένα τα λεγόμενα AJAX requests γράφονται σε JavaScript, και συνεπώς εκτελούνται στον περιηγητή του χρήστη.

Στην παρούσα Web εφαρμογή η τεχνολογία αυτή χρησιμοποιήθηκε έτσι ώστε να υπάρχει δυναμική ανανέωση των περιεχομένων των πινάκων της βάσης δεδομένων, και συνεπώς να μην είναι υποχρεωμένος ο χρήστης να ανανεώνει συνεχώς την σελίδα. Ακόμη, ανανεώνεται δυναμικά το κεντρικό Master Page της ιστοσελίδας, όπου εμφανίζονται όλες οι ειδοποιήσεις (notifications) προς τον χρήστη.

Η jQuery είναι ένα σύνολο από βιβλιοθήκες της γλώσσας JavaScript, οι οποίες ενσωματώνουν λειτουργίες που είναι πολύ πιθανό να θέλει κάποιος προγραμματιστής Web εφαρμογών να συμπεριλάβει στην ιστοσελίδα του. Είναι γραμμένες με αποδοτικό τρόπο, ευανάγνωστες, συντηρούνται και ανανεώνονται συνεχώς από την πολύ ενεργή κοινότητα που την αναπτύσσει. Ο σκοπός της είναι να κάνει την πλοήγηση στις ιστοσελίδες πιο ευχάριστη και διαδραστική γενικά να παρέχει μία ευχάριστη εμπειρία στον χρήστη μέσα από αυτήν. Ακόμη, παρέχει την δυνατότητα τροποποίησης των html κόμβων της ιστοσελίδας με εύκολο τρόπο, ενσωματώνοντας συντομογραφίες, συναρτήσεις και interfaces.

Στην παρούσα Web εφαρμογή η jQuery χρησιμοποιήθηκε για την εναλλαγή και παρουσίαση εικόνων στην αρχική σελίδα, καθώς επίσης και για την επαλήθευση (validation) των δεδομένων που εισάγει ο χρήστης. Η επαλήθευση αυτή είναι κομβικής σημασίας, αφού αφ ενός απαγορεύουμε την έλευση λανθασμένων δεδομένων στον Server (ακόμα και όταν κάνει submit ο χρήστης τα δεδομένα δεν αποστέλλονται αν είναι λανθασμένα), και εμφανίζεται κατάλληλο ενημερωτικό μήνυμα στον χρήστη ώστε να εισάγει σωστά δεδομένα, και αφ ετέρου δεν περιμένει ο χρήστης να αποσταλούν τα δεδομένα στον Server για να γίνει το validation, και μετά να απαντήσει ο Server με κάποιο μήνυμα λάθους. Βλέπουμε δηλαδή ότι η προσέγγιση αυτή είναι πολύ πιο αποδοτική.

4.6 CSS

Η CSS (Cascading Style Sheets) [9] είναι μια γλώσσα που επιτρέπει την εύκολη μορφοποίηση του περιεχομένου (της HTML δηλαδή) των ιστοσελίδων. Η γλώσσα αυτή αναπτύχθηκε από την διεθνή κοινότητα W3C [10] (World Wide Web Consortium) , και σε σύντομο χρονικό διάστημα προτυποποιήθηκε και υποστηρίζεται από όλους τους μοντέρνους περιηγητές.

Η ιδέα βάσει της οποίας αναπτύχθηκε αυτή η γλώσσα, είναι ο διαχωρισμός του περιεχομένου των ιστοσελίδων από την εμφάνισή τους. Ο διαχωρισμός αυτός επιτρέπει την συγγραφή κώδικα ο οποίος καθορίζει την εμφάνιση των ιστοσελίδων ξεχωριστά από το περιεχόμενό τους, καθιστώντας δυνατή την παράλληλη ανάπτυξη και των δύο από διαφορετικούς προγραμματιστές. Ακόμη είναι δυνατή η συγγραφή των λεγόμενων templates, που είναι κώδικας CSS σε κάποιο αρχείο, και η χρήση τους σε διαφορετικές ιστοσελίδες.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε η CSS, μέσω του αρχείου Site.css έτσι ώστε η μορφοποίηση που επιβάλλει να υιοθετείται από όλα τα Views της εφαρμογής, και να κάνει πλοήγηση στον ιστότοπο ευχάριστη.

4.7 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

4.7.1 Εργαλεία ανάπτυξης

Η Βάση Δεδομένων του πληροφοριακού συστήματος αναπτύχθηκε με τον Microsoft SQL Server 2010. Ο Microsoft SQL Server είναι ένα σχεσιακό μοντέλο διαχείρισης βάσεων δεδομένων (RDBMS) του οποίου η υλοποίηση της γλώσσας SQL ονομάζεται T-SQL. Οι λόγοι που χρησιμοποιούμε τα συστήματα διαχείρισης βάσεων δεδομένων είναι επειδή προσφέρουν [11]:

- *Ανεξαρτησία δεδομένων.* Τα δεδομένα είναι ανεξάρτητα από την εφαρμογή που τα χρησιμοποιεί. Μπορούν να δημιουργηθούν εφαρμογές οι οποίες να προσαρμόζονται με σχετική ευκολία σε διαφορετικές πηγές δεδομένων.
- *Ακεραιότητα δεδομένων.* Εξασφαλίζεται ότι δεν υπάρχουν ασυνέπειες μεταξύ εγγραφών που αντιπροσωπεύουν την ίδια πληροφορία, τα δεδομένα

συμμορφώνονται με τους περιορισμούς που επιβάλλονται από το σχήμα της βάσης, και οι τιμές των μεταβλητών συμμορφώνονται με το πεδίο τιμών και τον τύπο δεδομένων που έχει οριστεί.

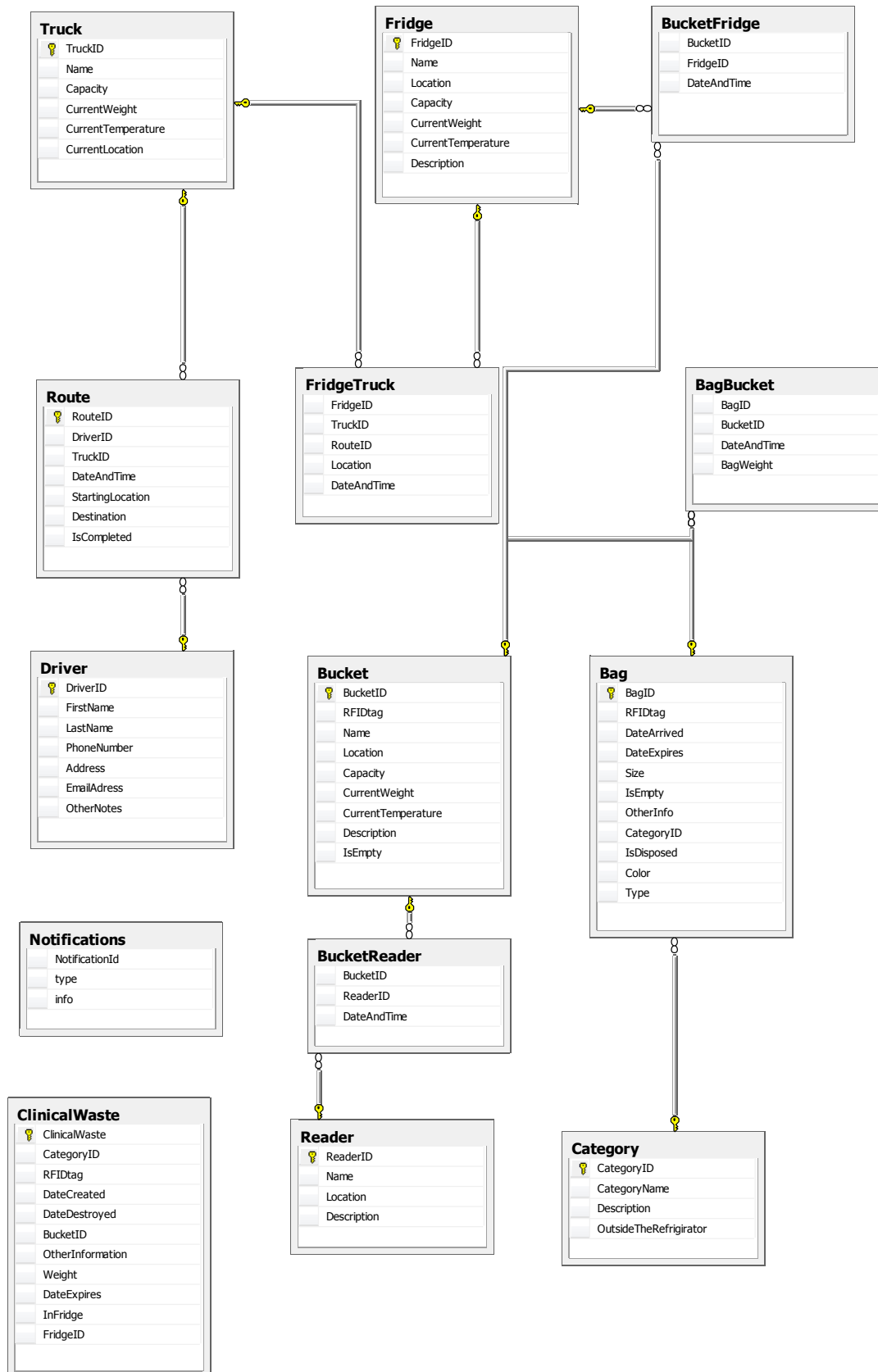
- *Ασφάλεια.* Εξασφαλίζει ότι η πρόσβαση στα δεδομένα είναι εξουσιοδοτημένη, μέσω των ελέγχων πρόσβασης που επιβάλλει.
- *Ταυτόχρονη προσπέλαση και κεντρικό έλεγχο.* Εξασφαλίζει την ταυτόχρονη προσπέλαση των δεδομένων και απαλείφει τα λεγόμενα αδιέξοδα (deadlocks).
- *Αποδοτική αποθήκευση και προσπέλαση των δεδομένων.* Χρησιμοποιεί αποδοτικά τον διαθέσιμο χώρο του υπολογιστικού συστήματος, και εισάγει δομές (ευρετήρια, hash tables κλπ) για να επιτυγχάνεται γρήγορη πρόσβαση στα δεδομένα
- *Αποφυγή πλεονάζουσας πληροφορίας (Non- Redundancy) των δεδομένων.* Το σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων επιχειρεί να εξασφαλίσει ότι δεν αποθηκεύεται πληροφορία η οποία είτε έχει αποθηκευτεί ήδη, είτε μπορεί να εξαχθεί από άλλες, ήδη αποθηκευμένες πληροφορίες.

Η δημιουργία και τροποποίηση της βάσης δεδομένων έγινε με την βοήθεια του SQL Server Management Studio. Το SQL Server Management studio είναι μία εφαρμογή με γραφικό περιβάλλον (GUI) και περιλαμβάνει όλα τα απαραίτητα εργαλεία για την επεξεργασία βάσεων δεδομένων, όπως script editors, object explorer και παρέχει αρκετές ευκολίες στον προγραμματιστή, όπως επισήμανση λαθών, Intellisense, αρκετό έτοιμο κώδικα κλπ.

4.7.2 Σχεσιακό Διάγραμμα της Βάσης Δεδομένων του Πληροφοριακού Συστήματος

Σε αυτή την ενότητα περιγράφεται το σχήμα της βάσης δεδομένων του πληροφοριακού συστήματος. Το σχήμα της βάσης προέκυψε από το μοντέλο οντοτήτων – συσχετίσεων που περιγράφηκε στην ενότητα 3.2. Να σημειωθεί σε αυτό το σημείο ότι δεν αντιστοιχίστηκαν όλες οι συσχετίσεις σε πίνακες του σχεσιακού μοντέλου, καθώς υπάρχει η δυνατότητα προσθήκης εξωτερικού κλειδιού (foreign key) στους πίνακες για τις σχέσεις πολλαπλότητας 1:N.

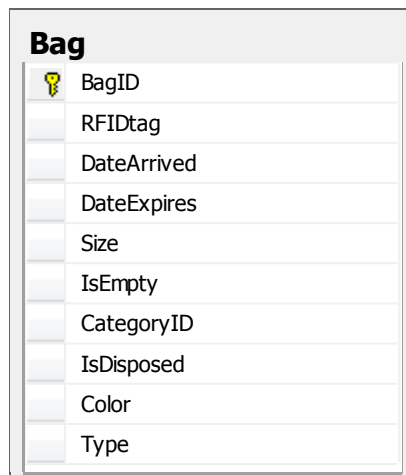
Ακολουθεί το διάγραμμα του σχήματος όπως αυτό προέκυψε από το SQL Server Management Studio, και μία σύντομη περιγραφή των πινάκων και των πεδίων τους.



Εικόνα 4.2. Σχεσιακό διάγραμμα της βάσης δεδομένων.

Οι πίνακες (σχέσεις) της βάσης δεδομένων είναι οι ακόλουθοι:

4.7.2.1 Table dbo.Bag



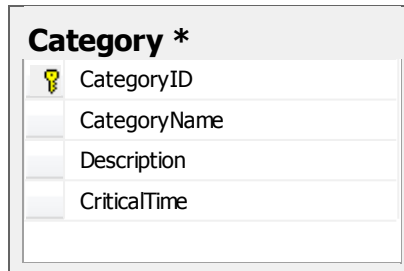
Bag	
🔑	BagID
	RFIDtag
	DateArrived
	DateExpires
	Size
	IsEmpty
	CategoryID
	IsDisposed
	Color
	Type


Εικόνα 4.3. Πίνακας dbo.Bag (Περιέκτης).

Αναπαριστά την οντότητα περιέκτες. Οι ιδιότητές του είναι οι ακόλουθες:

- *BagID (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα, ο αριθμός εκείνος δηλαδή που παραπέμπει σε μια συγκεκριμένη πλειάδα του (έναν περιέκτη). Δημιουργείται αυτόματα από το RDBMS με σκοπό την γρήγορη ανάκτηση μιας πλειάδας του πίνακα, και την γρήγορη συνένωση (join) με άλλους πίνακες.
- *RFID tag (uniqueidentifier)*. Είναι ένας unique identifier (Guid) που είναι μοναδικό σε ολόκληρη την βάση δεδομένων, και αντιπροσωπεύει το RFID tag που καταχωρείται στους περιέκτες.
- *DateArrived (datetime2)*. Είναι η ημερομηνία που καταχωρήθηκε ο περιέκτης στην βάση δεδομένων.
- *Size (int)*. Είναι ένας ακέραιος που υποδηλώνει το μέγεθος του περιέκτη.
- *IsEmpty (bit)*. Είναι ένα bit που παίρνει τις τιμές 1 για True και 0 για False, και υποδηλώνει την κατάσταση του περιέκτη (αν είναι άδειος ή όχι).
- *CategoryID (int)*. Είναι ένα εξωτερικό κλειδί (foreign key) που αναφέρεται στον πίνακα dbo.Categories, και υποδηλώνει την κατηγορία ιατρικών αποβλήτων που δύναται να φιλοξενήσει ο περιέκτης.
- *IsDisposed (bit)*. Είναι ένα bit που υποδηλώνει αν ο συγκεκριμένος περιέκτης έχει καταστραφεί.
- *Color (nvarchar(50))*. Είναι μία συμβολοσειρά που αντιστοιχεί στο χρώμα του περιέκτη.
- *Type (nvarchar(50))*. Είναι μια συμβολοσειρά που αντιστοιχεί στον τύπο του περιέκτη (plastic bag – hospital box).

4.7.2.2 Table dbo.Category




Category *	
	CategoryID
	CategoryName
	Description
	CriticalTime

Εικόνα 4.4. Πίνακας dbo.Category (Κατηγορία).

Αναπαριστά τις κατηγορίες ιατρικών αποβλήτων. Οι ιδιότητές του είναι οι ακόλουθες:

- *CategoryID* (*int*). Το πρωτεύον κλειδί (primary key) του πίνακα.
- *CategoryName* (*nvarchar(50)*). Είναι μια συμβολοσειρά που αντιστοιχεί στο όνομα της συγκεκριμένης κατηγορίας ιατρικών αποβλήτων.
- *Description* (*nvarchar(200)*). Είναι ένα μικρό κείμενο (*nvarchar(100)*) που αποτελεί μια σύντομη περιγραφή της συγκεκριμένης κατηγορίας ιατρικών αποβλήτων.
- *CriticalTime* (*decimal(10,1)*). Είναι ο μέγιστος χρόνος στον οποίο μπορούν τα απόβλητα του συγκεκριμένου τύπου να μείνουν στους συλλέκτες, δηλαδή εκτός των χώρων προσωρινής αποθήκευσης της νοσοκομειακής μονάδας.

4.7.2.3 Table dbo.Bucket


Bucket	
	BucketID
	RFIDtag
	Name
	Location
	Capacity
	CurrentWeight
	CurrentTemperature
	IsEmpty
	Description

Εικόνα 4.5. Πίνακας dbo.Bucket (Συλλέκτης).

Αναπαριστά την οντότητα συλλέκτες. Οι ιδιότητές του είναι οι ακόλουθες:

- *BucketID (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα.
- *RFIDtag (uniqueidentifier)*. Αντιπροσωπεύει το RFID tag που καταχωρείται στους συλλέκτες.
- *Name (nvarchar(50))*. Αντιστοιχεί στο όνομα του συγκεκριμένου συλλέκτη.
- *Location (nvarchar(100))*. Αντιστοιχεί στην τοποθεσία που βρίσκεται ο συγκεκριμένος συλλέκτης.
- *Capacity (int)*. Αντιστοιχεί στην προτεινόμενη μέγιστη χωρητικότητα (μετρούμενη σε κιλά) του συγκεκριμένου συλλέκτη.
- *CurrentWeight (decimal(10,1))*. Αντιπροσωπεύει το τρέχων βάρος (σε κιλά) των περιεκτών που περιέχει ο συλλέκτης.
- *CurrentTemperature (decimal(10,1))*. Αντιπροσωπεύει την θερμοκρασία στο περιβάλλον του συλλέκτη, μετρούμενη σε βαθμούς Κελσίου.
- *IsEmpty (bit)*. Εκφράζει την κατάσταση του συλλέκτη (αν είναι άδειος ή γεμάτος).
- *Description (nvarchar(50))*. Αντιστοιχεί στον τύπο του συλλέκτη (bin – trolley – cart).

4.7.2.4 Table dbo.Fridge


Fridge	
	FridgeID
	Name
	Location
	Capacity
	CurrentWeight
	CurrentTemperature
	Description

Εικόνα 4.6. Πίνακας dbo.Fridge (Χώροι προσωρινής αποθήκευσης).

Αναπαριστά την οντότητα χώροι προσωρινής αποθήκευσης. Οι ιδιότητές του είναι οι ακόλουθες:

- *FridgeID* (*int*). Το πρωτεύον κλειδί (primary key) του πίνακα.
- *Name* (*nvarchar(50)*). Αντιστοιχεί στο όνομα του συγκεκριμένου χώρου προσωρινής αποθήκευσης.
- *Location* (*nvarchar(50)*). Αντιστοιχεί στην τοποθεσία του συγκεκριμένου χώρου προσωρινής αποθήκευσης.
- *Capacity* (*int*). Αντιστοιχεί στην προτεινόμενη μέγιστη χωρητικότητα του.
- *CurrentWeight* (*decimal(10,1)*). Αντιστοιχεί στο καθαρό βάρος των απορριμμάτων που βρίσκονται στον χώρο προσωρινής αποθήκευσης.
- *CurrentTemperature* (*decimal(10,1)*). Αντιστοιχεί στην θερμοκρασία που επικρατεί στο περιβάλλον του.
- *Description* (*nvarchar(100)*). Αντιστοιχεί στον τύπο του χώρου προσωρινής αποθήκευσης (refrigerator – storeroom – lockable cupboard).

4.7.2.5 Table dbo.ClinicalWaste


ClinicalWaste	
	ClinicalWaste
	CategoryID
	RFIDtag
	DateCreated
	DateDestroyed
	BucketID
	Weight
	DateExpires
	InFridge
	FridgeID

Εικόνα 4.7. Πίνακας dbo.ClinicalWaste

Αναπαριστά τα ιατρικά απόβλητα που δημιουργήθηκαν εντός της νοσοκομειακής μονάδας. Οι ιδιότητες του είναι οι ακόλουθες:

- *ClinicalWaste (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα.
- *CategoryID (int)*. Εξωτερικό κλειδί (foreign key) που παραπέμπει σε μία πλειάδα του πίνακα dbo.Category.
- *RFIDtag (uniqueidentifier)*. Αντιπροσωπεύει το RFID tag του περιέκτη που φιλοξένησε το απόβλητο.
- *DateCreated (datetime2)*. Αντιπροσωπεύει την ημερομηνία δημιουργίας του.
- *DateDestroyed (datetime2)*. Αντιπροσωπεύει την ημερομηνία καταστροφής του συγκεκριμένου περιέκτη (αν έχει καταστραφεί).
- *BucketID (int)*. Εξωτερικό κλειδί (foreign key) που παραπέμπει σε μια πλειάδα του πίνακα dbo.Bucket, και υποδηλώνει τον συλλέκτη στον οποίο δημιουργήθηκε το απόβλητο.
- *Weight (decimal(10,1))*. Υποδηλώνει το βάρος του συγκεκριμένου περιέκτη.
- *DateExpires (datetime2)*. Υποδηλώνει την ώρα και ημερομηνία μέχρι την οποία μπορούν τα απόβλητα του συγκεκριμένου τύπου να μείνουν στους συλλέκτες, δηλαδή εκτός των χώρων προσωρινής αποθήκευσης της νοσοκομειακής μονάδας.
- *InFridge (bit)*. Δείχνει αν ο συγκεκριμένος (γεμάτος) περιέκτης είναι εντός κάποιου χώρου προσωρινής αποθήκευσης.
- *FridgeID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μια πλειάδα του πίνακα dbo.Fridge, και υποδηλώνει τον χώρο προσωρινής αποθήκευσης μέσα στον οποίο βρίσκεται ο συγκεκριμένος περιέκτης.

4.7.2.6 Table dbo.Reader


Reader	
	ReaderID
	Name
	Location
	Description

Εικόνα 4.8. Πίνακας dbo.Reader (Αναγνώστης)

Αναπαριστά την οντότητα αναγνώστες (readers). Οι ιδιότητές του είναι οι ακόλουθες:

- *ReaderID (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα.
- *Name (nvarchar(50))*. Αναπαριστά το όνομα του συγκεκριμένου αναγνώστη.
- *Location (nvarchar(100))*. Αντιστοιχεί την τοποθεσία στην οποία βρίσκεται ο συγκεκριμένος αναγνώστης

4.7.2.7 Table dbo.Truck

Truck	
	TruckID
	Name
	Capacity
	CurrentWeight
	CurrentTemperature
	CurrentLocation

Εικόνα 4.9. Πίνακας dbo.Truck (Οχήματα)

Αναπαριστά τα οχήματα που είναι υπεύθυνα για την μεταφορά των απορριμμάτων από τους χώρους προσωρινής αποθήκευσης της νοσοκομειακής μονάδας στον χώρο διαχείρισής τους. Οι ιδιότητές του είναι οι ακόλουθες:

- *TruckID (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα.
- *Name (nvarchar(50))*. Το όνομα (μοντέλο) του συγκεκριμένου οχήματος.
- *Capacity (decimal(10,1))*. Αναπαριστά το μέγιστο ονομαστικό βάρος του οχήματος.

- *CurrentTemperature (decimal(10,1))*. Αναπαριστά την θερμοκρασία που επικρατεί στον χώρο όπου βρίσκονται τα απορρίμματα.
- *CurrentLocation (geodata)*. Αναπαριστά την τρέχουσα τοποθεσία του οχήματος.

4.7.2.8 Table BagBucket

BagBucket	
	BagID
	BucketID
	DateAndTime
	BagWeight

Εικόνα 4.10. Πίνακας dbo.BagBucket

Αντιπροσωπεύει την συσχέτιση δέχτηκε, μεταξύ του συλλέκτη και του περιέκτη ιατρικών αποβλήτων. Οι ιδιότητές του είναι οι ακόλουθες:

- *BagID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μία εγγραφή του πίνακα dbo.Bag. Αντιπροσωπεύει δηλαδή έναν περιέκτη.
- *BucketID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μια εγγραφή του πίνακα dbo.Bucket. Αντιπροσωπεύει δηλαδή έναν συλλέκτη.
- *DateandTime (datetime2)*. Αντιστοιχεί στην ώρα και ημερομηνία κατά την οποία καταγράφηκε ο συγκεκριμένος περιέκτης από τον συγκεκριμένο συλλέκτη.
- *BagWeight (decimal (10,1))*. Αντιστοιχεί στο βάρος του περιέκτη.

4.7.2.9 Table BucketFridge

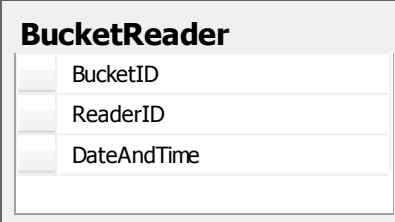
BucketFridge	
	BucketID
	FridgeID
	DateAndTime

Εικόνα 4.11. Πίνακας dbo.BucketFridge

Αντιπροσωπεύει την συσχέτιση εκκενώθηκε σε, μεταξύ ενός συλλέκτη και ενός χώρου προσωρινής αποθήκευσης. Οι ιδιότητές του είναι οι ακόλουθες:

- *BucketID* (*int*). Εξωτερικό κλειδί (*foreign key*), που αναφέρεται σε έναν συλλέκτη.
- *FridgeID* (*int*). Εξωτερικό κλειδί (*foreign key*), που αναφέρεται σε έναν χώρο προσωρινής αποθήκευσης.
- *DateAndTime* (*datetime2*). Αντιπροσωπεύει την ώρα και ημερομηνία κατά την οποία εκκενώθηκε ο συγκεκριμένος συλλέκτης στον συγκεκριμένο χώρο προσωρινής αποθήκευσης.

4.7.2.10 Table **dbo.BucketReader**



BucketReader	
	BucketID
	ReaderID
	DateAndTime

Εικόνα 4.12. Πίνακας **dbo.BucketReader**

Αντιπροσωπεύει την συσχέτιση καταγράφηκε από, μεταξύ ενός συλλέκτη και ενός αναγνώστη (*reader*). Οι ιδιότητές του είναι οι ακόλουθες:

- *BucketID* (*int*). Εξωτερικό κλειδί (*foreign key*) που αναφέρεται σε έναν συλλέκτη.
- *ReaderID* (*int*). Εξωτερικό κλειδί (*foreign key*) που αναφέρεται σε έναν αναγνώστη.
- *DateAndTime* (*datetime2*). Αντιστοιχεί στην ώρα και ημερομηνία που καταγράφηκε ο συγκεκριμένος συλλέκτης από τον συγκεκριμένο αναγνώστη.

4.7.2.11 Table **dbo.FridgeTruck**

FridgeTruck	
	FridgeID
	TruckID
	RouteID
	Location
	DateAndTime

Εικόνα 4.13. Πίνακας dbo.FridgeTruck.

Αναπαριστά την συσχέτιση φορτώθηκε σε, μεταξύ ενός χώρου προσωρινής αποθήκευσης και ενός οχήματος μεταφοράς. Οι ιδιότητές του είναι οι ακόλουθες:

- *FridgeID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε έναν χώρο προσωρινής αποθήκευσης.
- *TruckID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε ένα όχημα μεταφοράς
- *RouteID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μία πλειάδα του πίνακα dbo.Route.
- *Location (geolodata)*. Αναφέρεται στην τοποθεσία φόρτωσης των απορριμμάτων.
- *DateAndTime (datetime2)*. Αντιστοιχεί στην ώρα και ημερομηνία της φόρτωσης.

4.7.2.12 Table dbo.Route

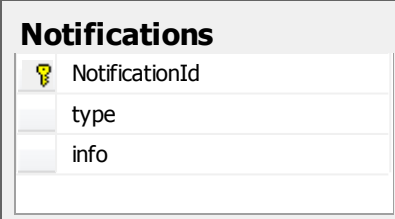
Route	
🔑	RouteID
	DriverID
	TruckID
	DateAndTime
	StartingLocation
	Destination
	IsCompleted


Εικόνα 4.14. Πίνακας dbo.Route

Αναπαριστά την συσχέτιση έκανε διαδρομή, μεταξύ του οδηγού και του οχήματος. Οι ιδιότητές του είναι οι ακόλουθες:

- *RouteID (int)*. Το πρωτεύον κλειδί (primary key) του πίνακα.
- *DriverID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μια πλειάδα του πίνακα Driver.
- *TruckID (int)*. Εξωτερικό κλειδί (foreign key) που αναφέρεται σε μια πλειάδα του πίνακα Truck.
- *DateAndTime (datetime2)*. Αναπαριστά την ημερομηνία που πραγματοποιήθηκε το συγκεκριμένο δρομολόγιο.
- *StartingLocation (geodata)*. Αναπαριστά την τοποθεσία αφετηρίας του δρομολογίου.
- *Destination (geodata)*. Αναπαριστά τον προορισμό του δρομολογίου.
- *IsCompleted (bit)*. Δείχνει εάν το συγκεκριμένο δρομολόγιο έχει ολοκληρωθεί ή όχι.

4.7.2.13 Table dbo.Notifications



Notifications	
	NotificationId
	type
	info

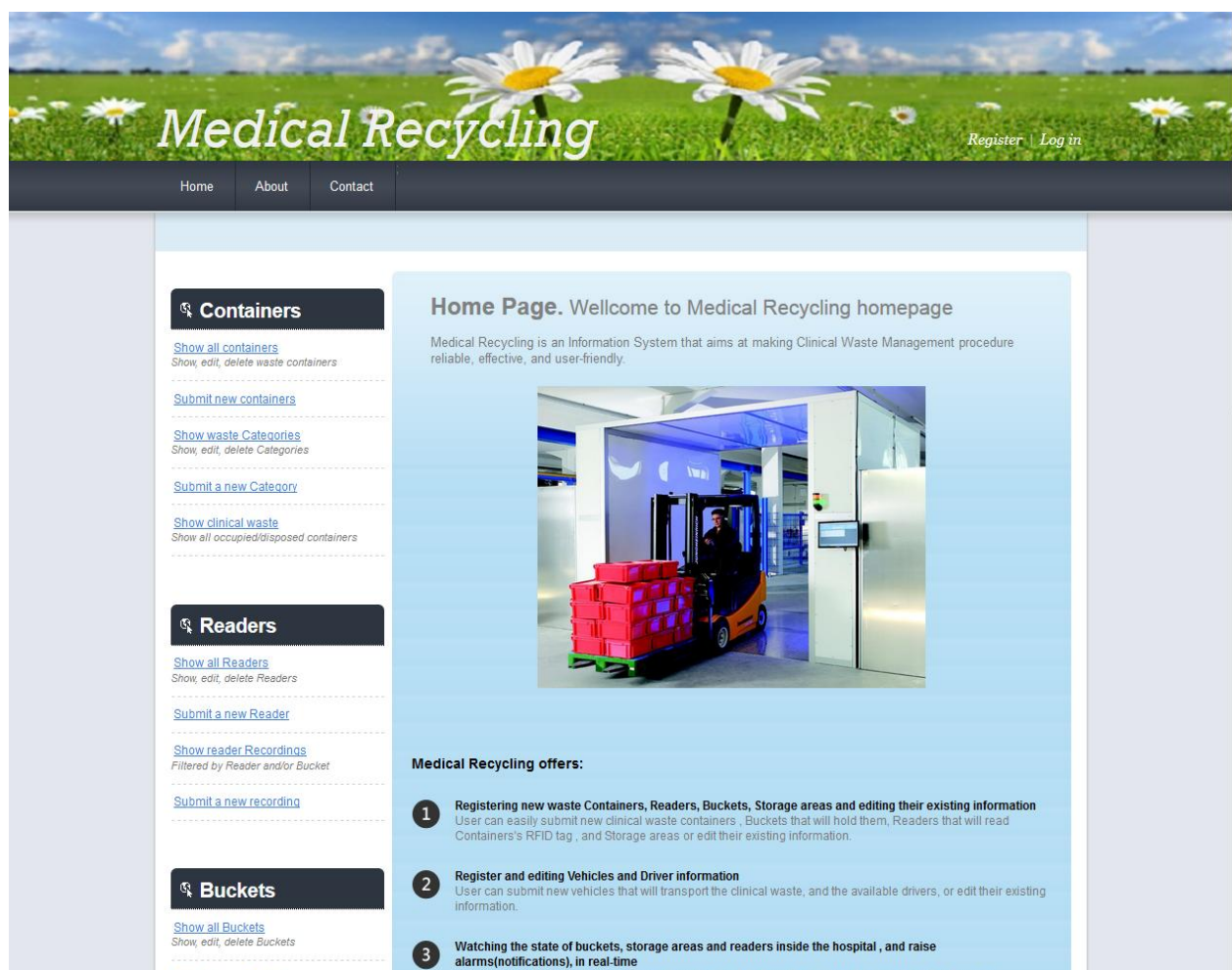
Εικόνα 4.15. Πίνακας dbo.Notifications

Αναπαριστά τις ειδοποιήσεις που εμφανίζονται στον χρήστη, εάν παραβιάζεται κάποιος από τους κανόνες λειτουργίας. Οι ιδιότητές του είναι:

- *NotificaionID (int)*. Πρωτεύον κλειδί (primary key) του πίνακα.
- *Type (nvarchar(50))*. Αντιπροσωπεύει τον τύπο της ειδοποίησης.
- *Info (nvarchar(200))*. Αναπαριστά όλες τις απαραίτητες λεπτομέρειες για την συγκεκριμένη ειδοποίηση.

5 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ

Σε αυτό το κεφάλαιο παρουσιάζεται το ολοκληρωμένο πληροφοριακό σύστημα μέσα από παραδείγματα χρήσης. Στην παρακάτω εικόνα φαίνεται η αρχική σελίδα της εφαρμογής μας.



Εικόνα 5.1. Η αρχική σελίδα της εφαρμογής.

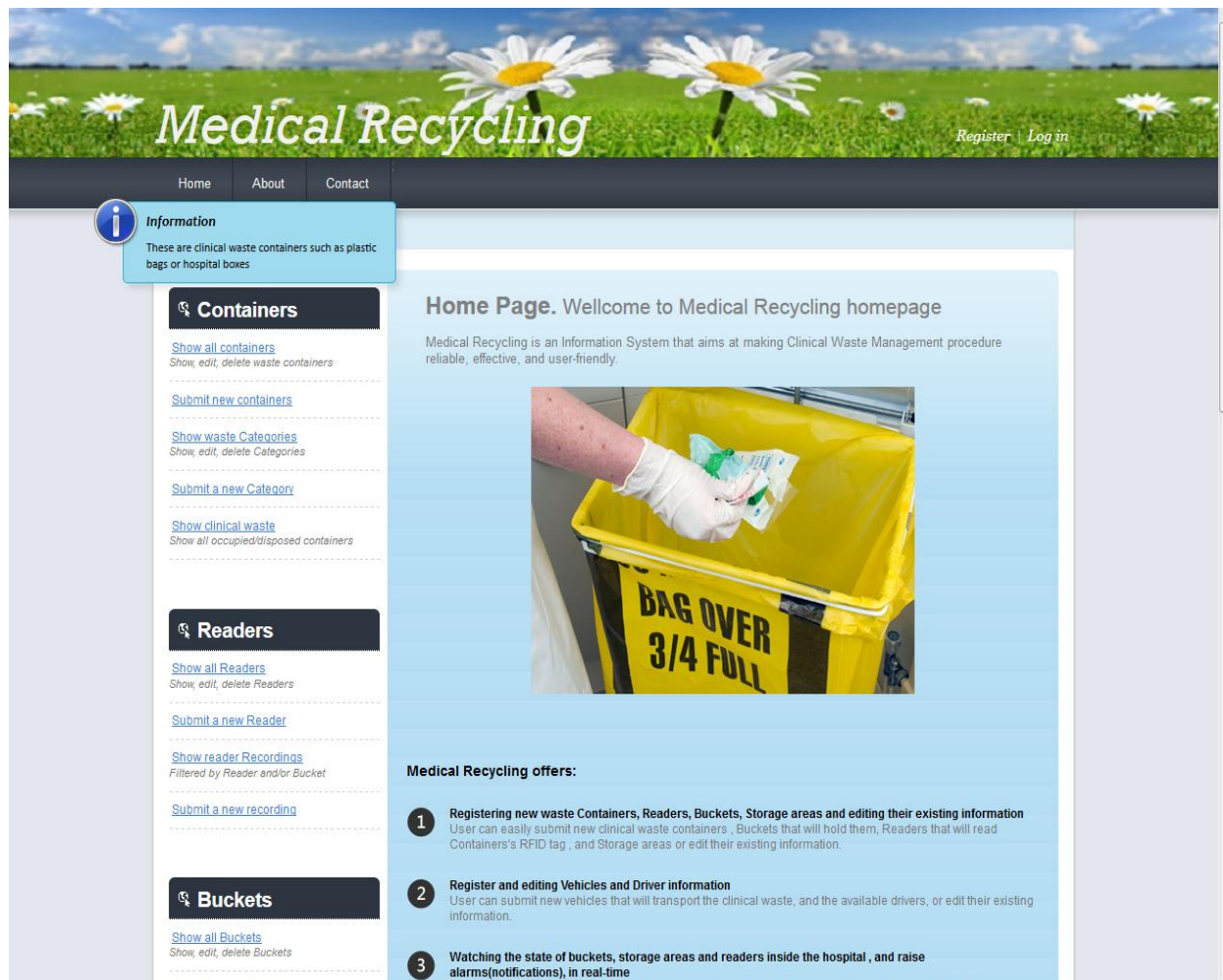
Βλέπουμε κατ αρχήν ότι υπάρχει ένα κύριο menu bar στην κορυφή με τα Links Home, About, Contact (αργότερα θα δούμε ότι εμφανίζονται και τα notifications σε εκείνο το σημείο, όταν ο χρήστης κάνει log in και υπάρχουν ειδοποιήσεις), ένα sidebar στο αριστερό μέρος της ιστοσελίδας, που επιτρέπει την πλοήγηση στην εφαρμογή. Στο πάνω μέρος της ιστοσελίδας φαίνεται το λογότυπο της εφαρμογής, και η εικόνα που χρησιμοποιήθηκε για background image. Αυτά τα στοιχεία αποτελούν το Master Page που θα εμφανίζεται σε όλα τα Views της εφαρμογής.

Στο κεντρικό μέρος της ιστοσελίδας εμφανίζεται το περιεχόμενο του κάθε View. Στην παραπάνω εικόνα βλέπουμε το View που εμφανίζεται σε έναν χρήστη ο οποίος δεν έχει κάνει log in. Προβάλλονται πληροφορίες σχετικά με την λειτουργικότητα που παρέχει η εφαρμογή, στο επάνω δεξιά μέρος βρίσκονται τα link που μπορεί ο χρήστης να συνδεθεί (log in) ή να δημιουργήσει λογαριασμό (Register) στον ιστότοπο. Υπάρχει ακόμα ένα slideshow, όπου παρουσιάζονται εικόνες σχετικές με το θέμα του πληροφοριακού συστήματος, οι οποίες κάνουν cross fade μεταξύ τους (αλλάζουν κάθε 5 δευτερόλεπτα). Μετά από 5 δευτερόλεπτα δηλαδή η εικόνα που βλέπει ο χρήστης είναι η παρακάτω:



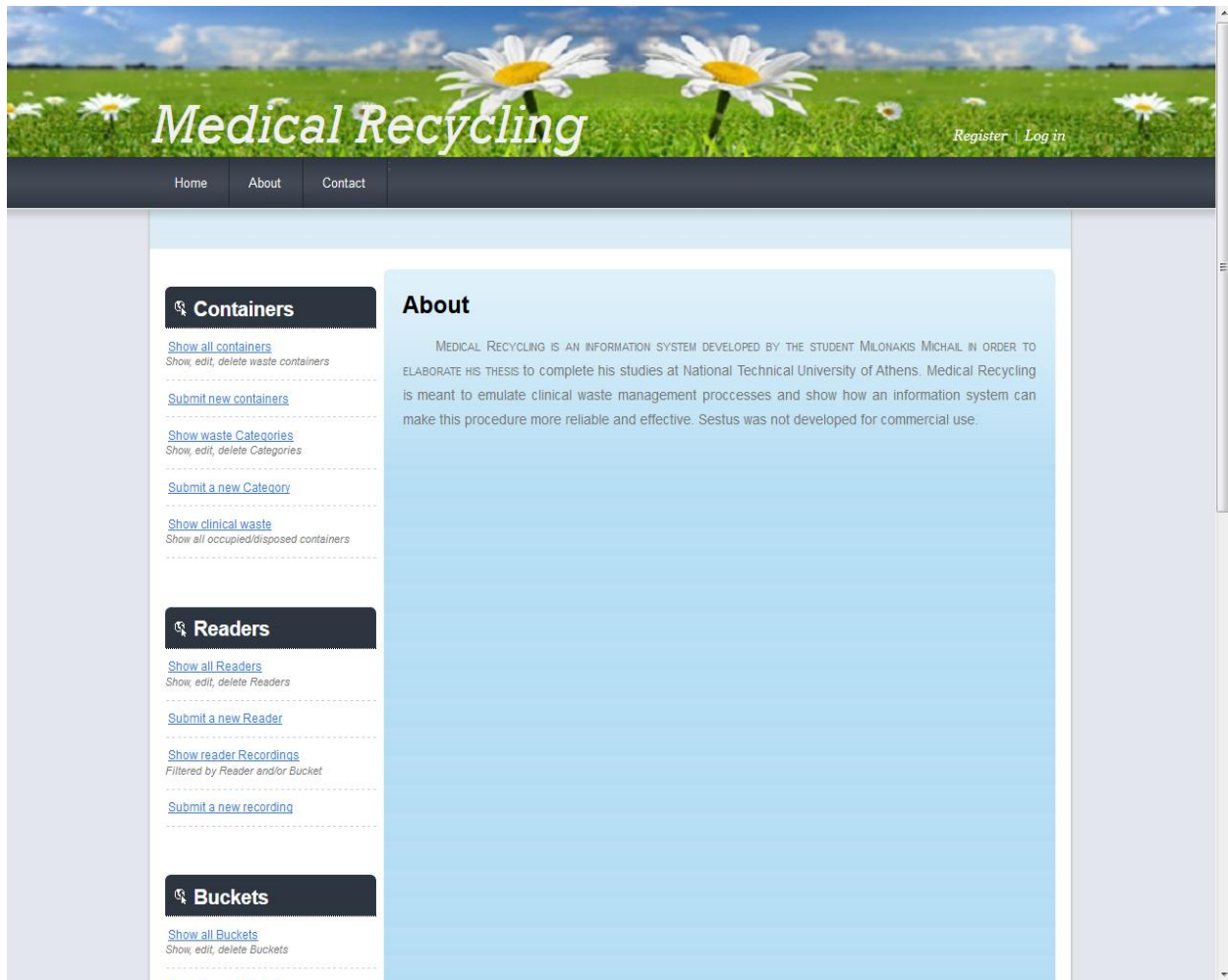
Εικόνα 5.2. Αρχική σελίδα.

Προσπαθώντας να παρέχουμε όσο το δυνατόν πιο ευχάριστη εμπειρία στον χρήστη που επισκέπτεται τον ιστότοπο, και να γίνει η ιστοσελίδα πιο διαδραστική, όταν ο χρήστης οδηγεί τον κέρσορα (hover) σε ένα από τα κύρια στοιχεία του Sidebar (Containers, Readers, Buckets, Storage Areas), εμφανίζεται ένα tooltip (ένα όμορφο γραφικό στοιχείο που περιέχει κείμενο) που εξηγεί τι είναι το συγκεκριμένο στοιχείο. Όταν για παράδειγμα κάνουμε hover στο αντικείμενο Containers του sidebar βλέπουμε την παρακάτω εικόνα:



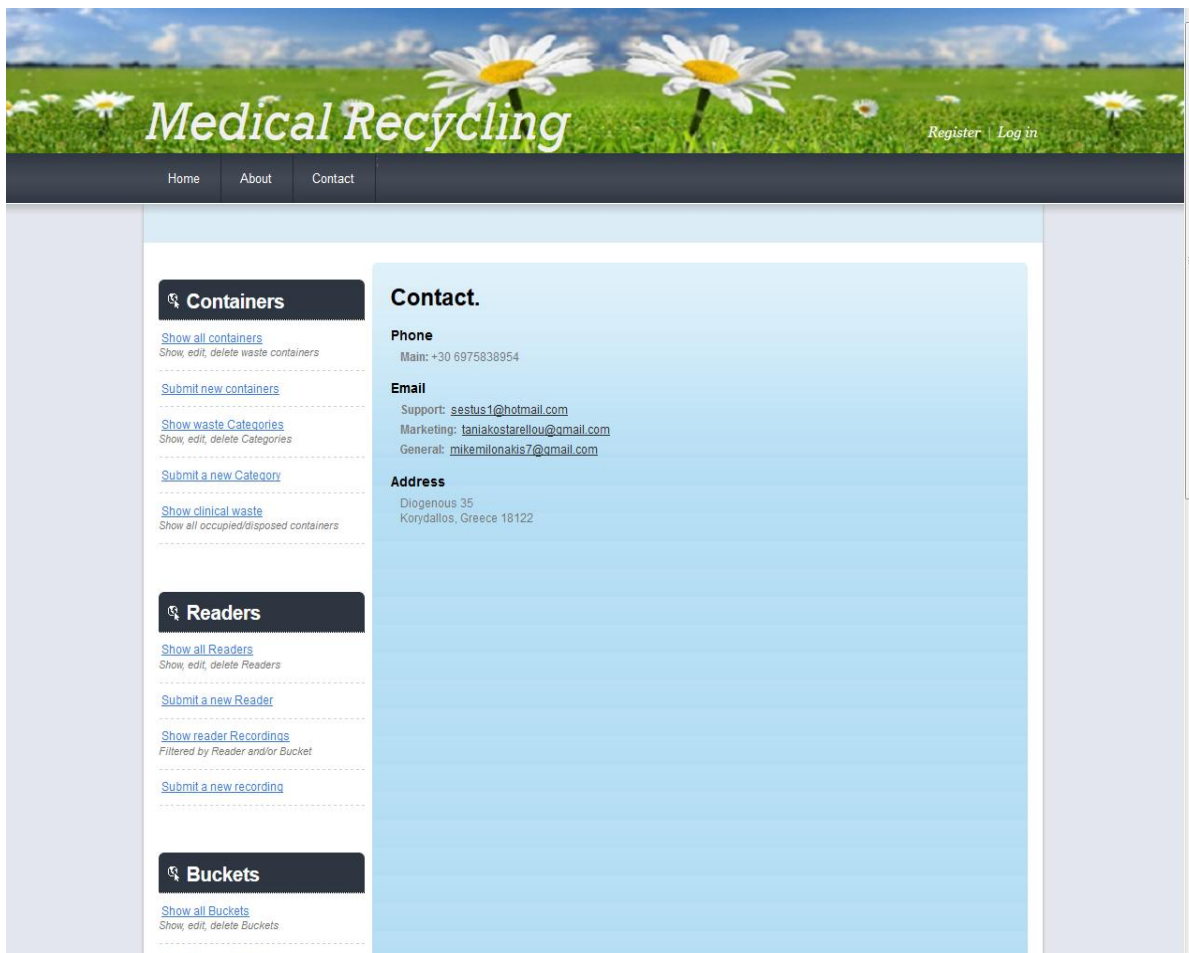
Εικόνα 5.3. Tooltip.

Αν ο χρήστης ακολουθήσει το link *About*, εμφανίζονται πληροφορίες για το παρόν πληροφοριακό σύστημα, όπως από ποιον αναπτύχθηκε, ο σκοπός του, και γενικά οποιαδήποτε άλλη πληροφορία θεωρείται χρήσιμη. Το συγκεκριμένο View είναι το παρακάτω:



Εικόνα 5.4. Η σελίδα About.

Το link *Contact* εμφανίζει τα στοιχεία επικοινωνίας με τους υπεύθυνους του ιστότοπου. Το συγκεκριμένο View είναι το ακόλουθο:



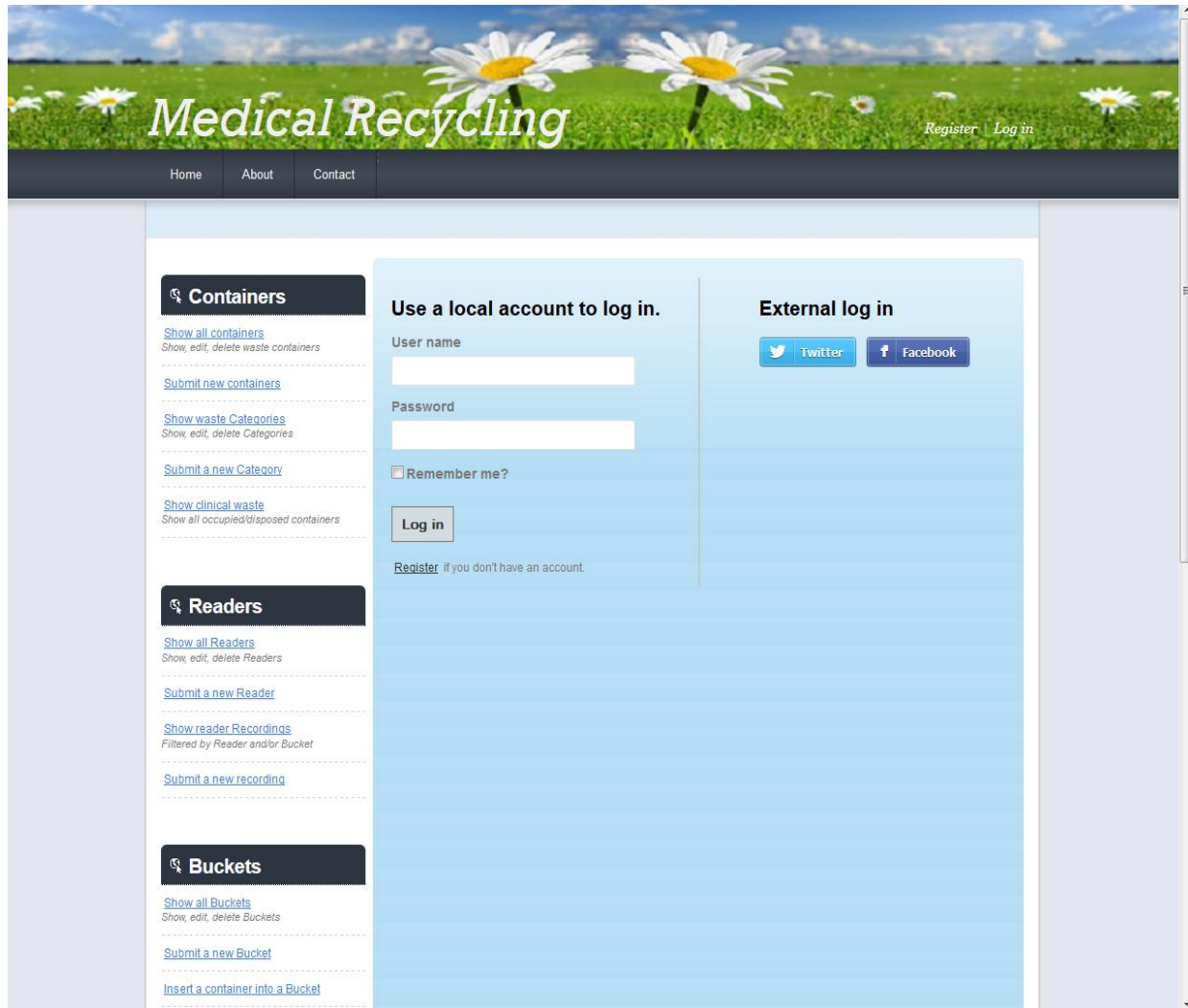
Εικόνα 5.5. Η σελίδα Contact.

Αν ο χρήστης επιχειρήσει να χρησιμοποιήσει κάποια από τις λειτουργίες που προσφέρει το πληροφοριακό σύστημα, όπως για παράδειγμα να δει όλους τους καταχωρημένους αναγνώστες της νοσοκομειακής μονάδας, θα του ζητηθεί να συνδεθεί στο σύστημα, έτσι ώστε να αποφευχθεί μη εξουσιοδοτημένη πρόσβαση στα δεδομένα. Ο χρήστης λοιπόν παραπέμπεται στο View όπου του ζητείται να συνδεθεί.

Ακολουθούν τα σενάρια για κάθε λειτουργικότητα που παρέχεται από το πληροφοριακό σύστημα, ξεκινώντας από την σύνδεση (log in) του χρήστη. Να σημειωθεί πως αρκετά από τα σενάρια που παρουσιάζονται, τυπικά θα συνέβαιναν αυτόματα σε μια νοσοκομειακή μονάδα που θα είχε τις απαραίτητες υποδομές, όπως για παράδειγμα η καταγραφή ενός συλλέκτη από έναν αναγνώστη (reader), είτε από τους εργαζόμενους (γιατρούς, νοσοκόμες) εντός αυτής, η εισαγωγή ενός περιέκτη σε έναν συλλέκτη, το άδειασμα ενός συλλέκτη σε κάποιον χώρο προσωρινής αποθήκευσης, η εκκένωση ενός χώρου προσωρινής αποθήκευσης σε κάποιο όχημα μεταφοράς κλπ. Επειδή δεν υπήρχε πρόσβαση σε τέτοιες υποδομές, όλες οι λειτουργικότητες που περιγράφηκαν γίνονται 'εικονικά' μέσα από την ιστοσελίδα.

5.1 Σύνδεση (Log in) Χρήστη

Σε αυτό το σενάριο ο χρήστης επιχειρεί να συνδεθεί στο πληροφοριακό σύστημα, ακολουθώντας το link στο επάνω – δεξιά μέρος της σελίδας. Ακολουθώντας το Link οδηγείται στην παρακάτω σελίδα:



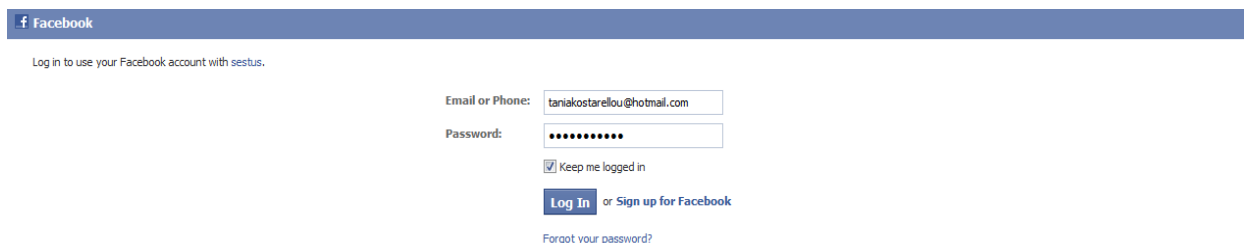
Εικόνα 5.6. Σελίδα Log in.

Βλέπουμε ότι ο χρήστης καλείται συμπληρώσει το όνομα χρήστη και τον κωδικό του, και προσφέρεται η δυνατότητα αποθήκευσης αυτών των στοιχείων (Remember me) έτσι ώστε να μην χρειάζεται ο χρήστης να τα συμπληρώνει, κάθε φορά που επισκέπτεται την σελίδα χρησιμοποιώντας τον ίδιο browser.

Μία ακόμη σημαντική λειτουργικότητα που υλοποιήθηκε είναι η δυνατότητα εισόδου μέσω *social media*, και συγκεκριμένα χρησιμοποιώντας το **Facebook** ή το **Twitter**.

Τα *social media* έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητας ενός αρκετά μεγάλου ποσοστού του πληθυσμού των ανεπτυγμένων χωρών. Γι αυτόν τον λόγο προσφέρεται η δυνατότητα σύνδεσης μέσω του λογαριασμού σε κάποιο από αυτά τα *social media*, χωρίς δηλαδή να χρειάζεται εγγραφή (και) στην συγκεκριμένη σελίδα. Αυτή η προσέγγιση είναι αρκετά βολική για τον χρήστη, καθώς δεν χρειάζεται να διατηρεί έναν ακόμη λογαριασμό, πιθανώς με διαφορετικά στοιχεία από άλλους που έχει ήδη δημιουργήσει σε άλλους ιστότοπους, αλλά συνδέεται με ένα *account* που έχει δημιουργήσει παλαιότερα για κάποιο από αυτά τα *social media*, και φυσικά η διαδικασία αυτή είναι πιο γρήγορη από το να χρειαζόταν καινούρια εγγραφή (*registration*).

Αν ο χρήστης επιλέξει να συνδεθεί για παράδειγμα μέσω του *Facebook* παραπέμπεται στους *servers* του, όπου ερωτάται αν επιτρέπει να συνδεθεί στην σελίδα μας μέσω του λογαριασμού του εκεί. Η σελίδα που βλέπει είναι η ακόλουθη:



f Facebook

Log in to use your Facebook account with sestus.

Email or Phone: taniakostarellou@hotmail.com

Password: ●●●●●●●●

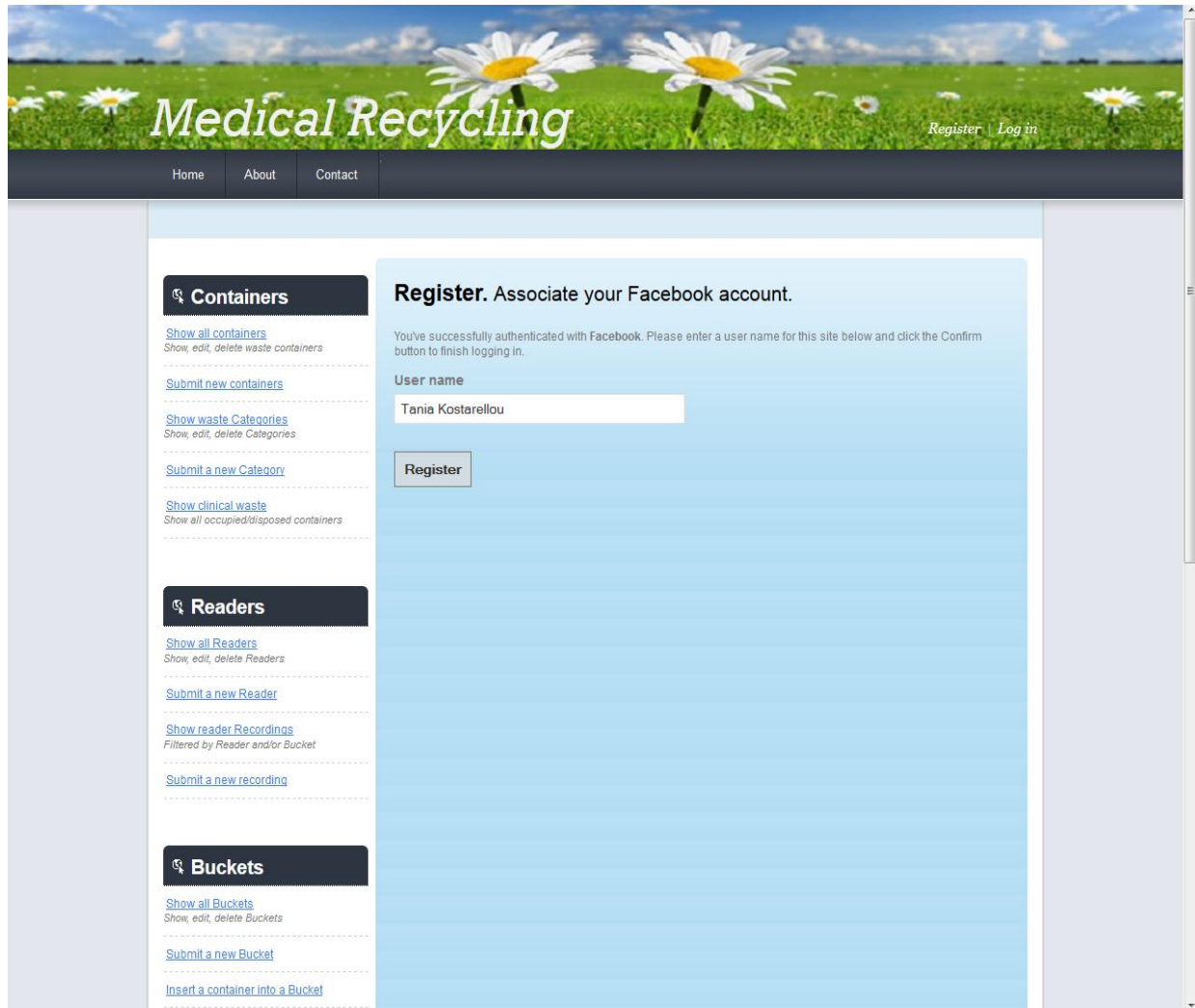
Keep me logged in

Log In or Sign up for Facebook

[Forgot your password?](#)

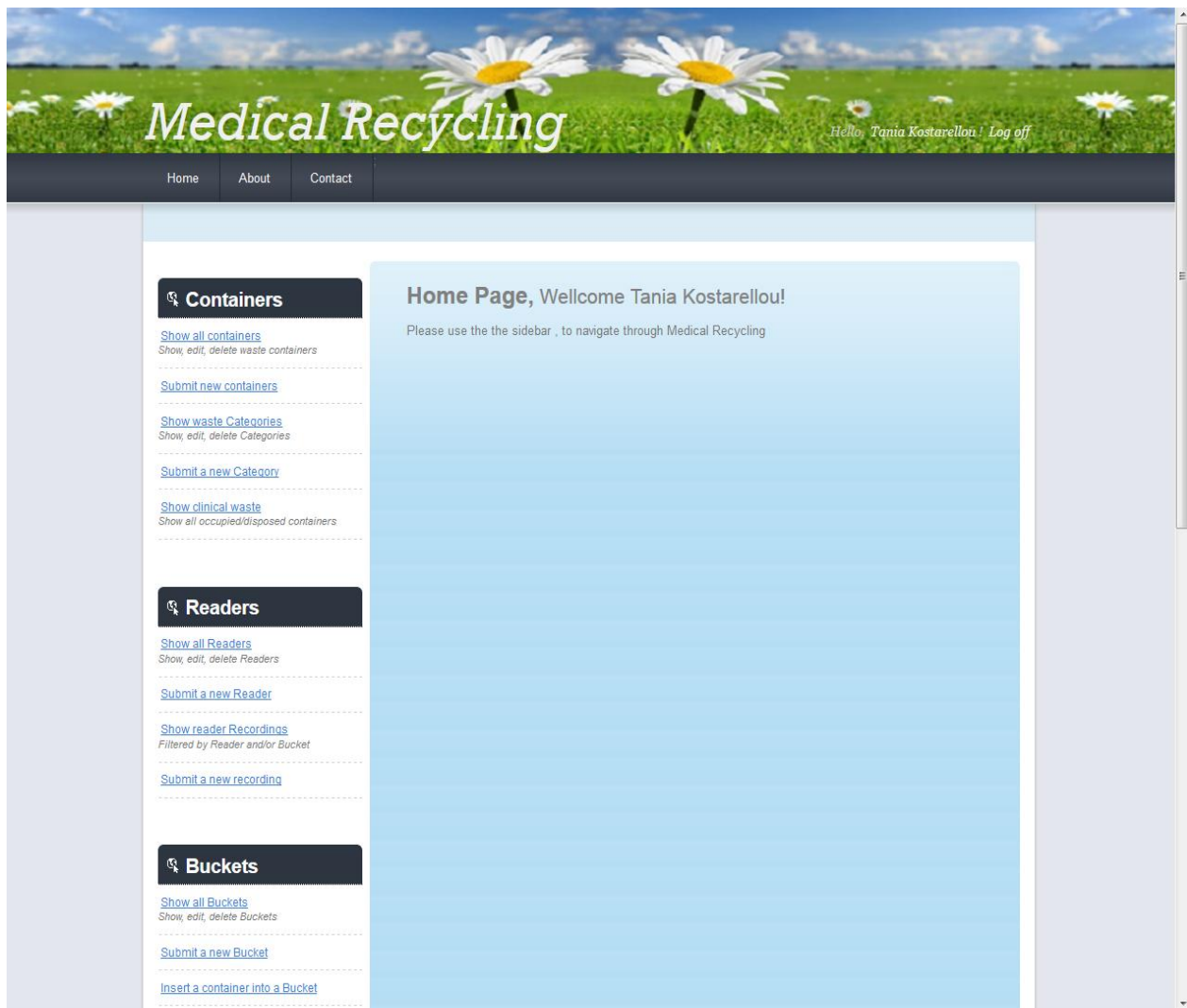
Εικόνα 5.7. Log in μέσω Facebook.

Αν ο χρήστης επιλέξει να συνδεθεί, τότε επιστρέφει στην ιστοσελίδα μας, όπου του εμφανίζεται κατάλληλο ενημερωτικό μήνυμα που αναφέρει πως έγινε επιτυχημένη είσοδος με το *Facebook* και απλά ζητείται ένα User name, το οποίο είναι by default ίδιο με το User name στο *Facebook*, που θα εμφανίζεται στον ιστότοπο.



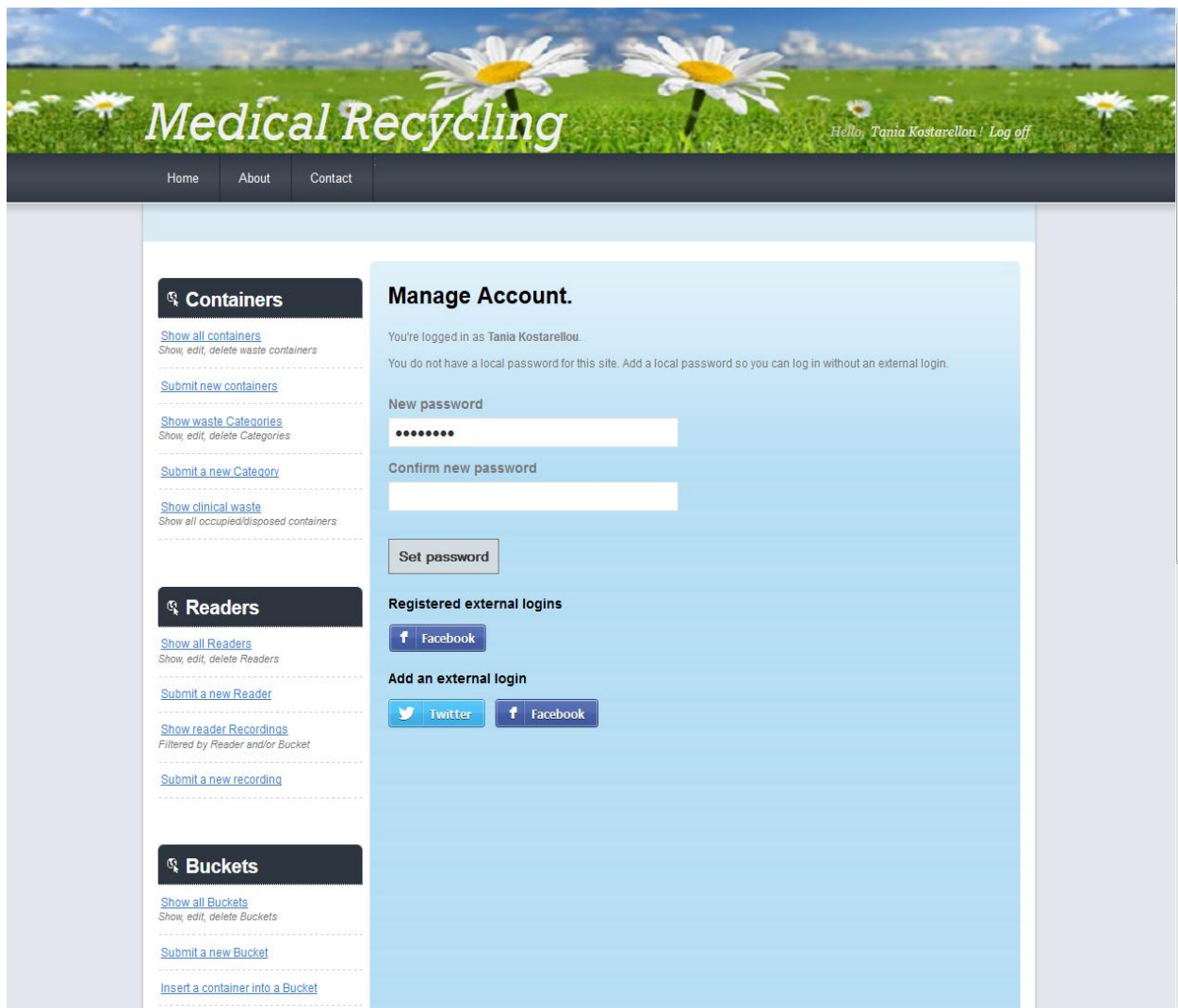
Εικόνα 5.8. Επιτυχημένη σύνδεση μέσω *Facebook*.

Αφού λοιπόν ο χρήστης επιλέξει το user name του, ή αφήσει το ήδη υπάρχων user name του στο *Facebook*, η σύνδεση ολοκληρώνεται και εμφανίζεται κατάλληλο ενημερωτικό μήνυμα, που τον καλωσορίζει στον ιστότοπο:



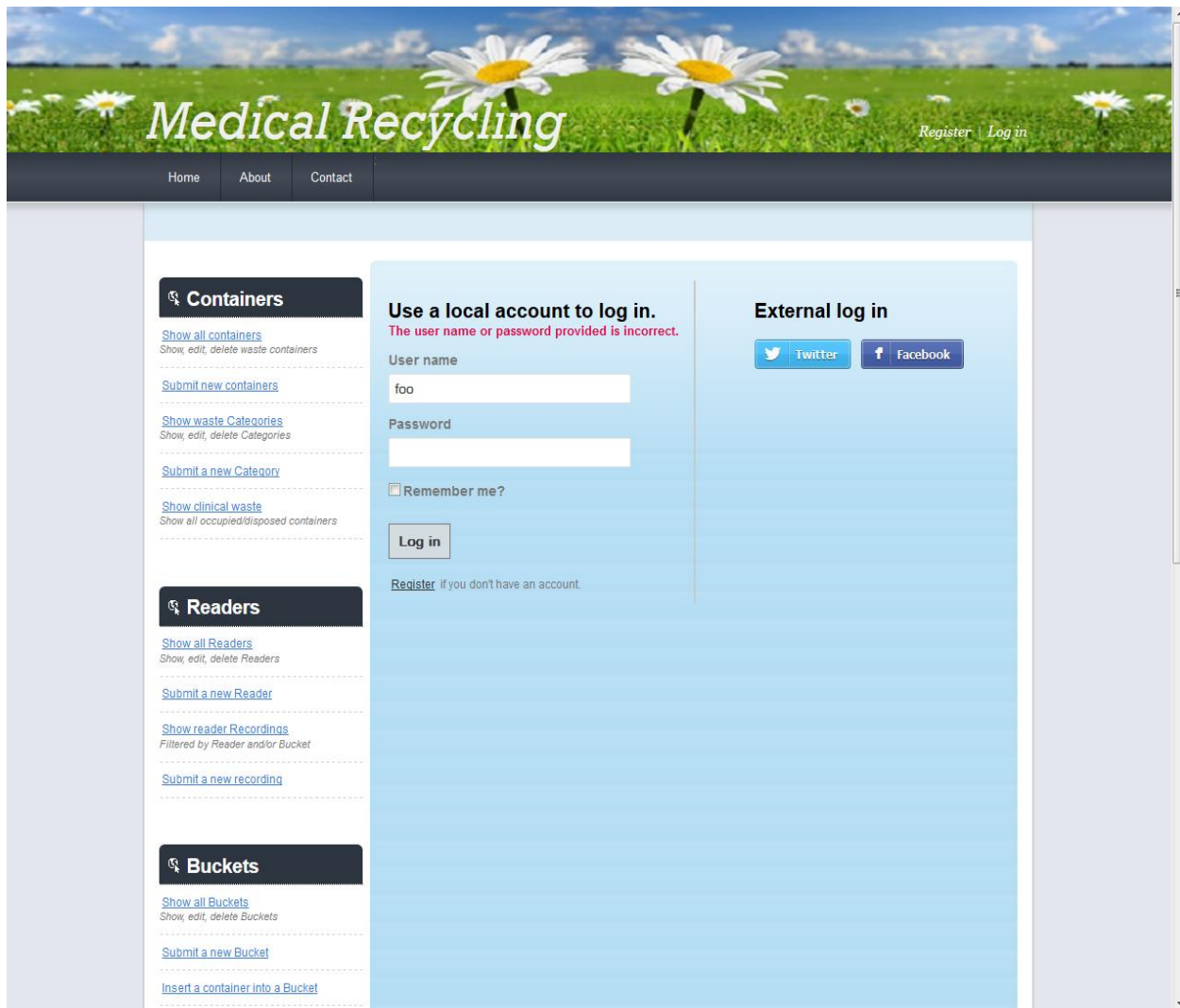
Εικόνα 5.9. Επιτυχής σύνδεση.

Βλέπουμε ότι στο επάνω δεξιά μέρος της σελίδας, όπου πριν βρίσκονταν τα links *Register / Log in*, τώρα εμφανίζεται ένα link το οποίο έχει το όνομα του χρήστη (της μορφής Hello, <<user name>>), το οποίο παραπέμπει στην σελίδα διαχείρισης του προφίλ (manage account), όπου υπάρχει η δυνατότητα αλλαγής του κωδικού πρόσβασης. Ακόμη, δίπλα από αυτό το Link, εμφανίζεται και το link για την αποσύνδεση (log off) του χρήστη.



Εικόνα 5.10. Σελίδα διαχείρισης του λογαριασμού.

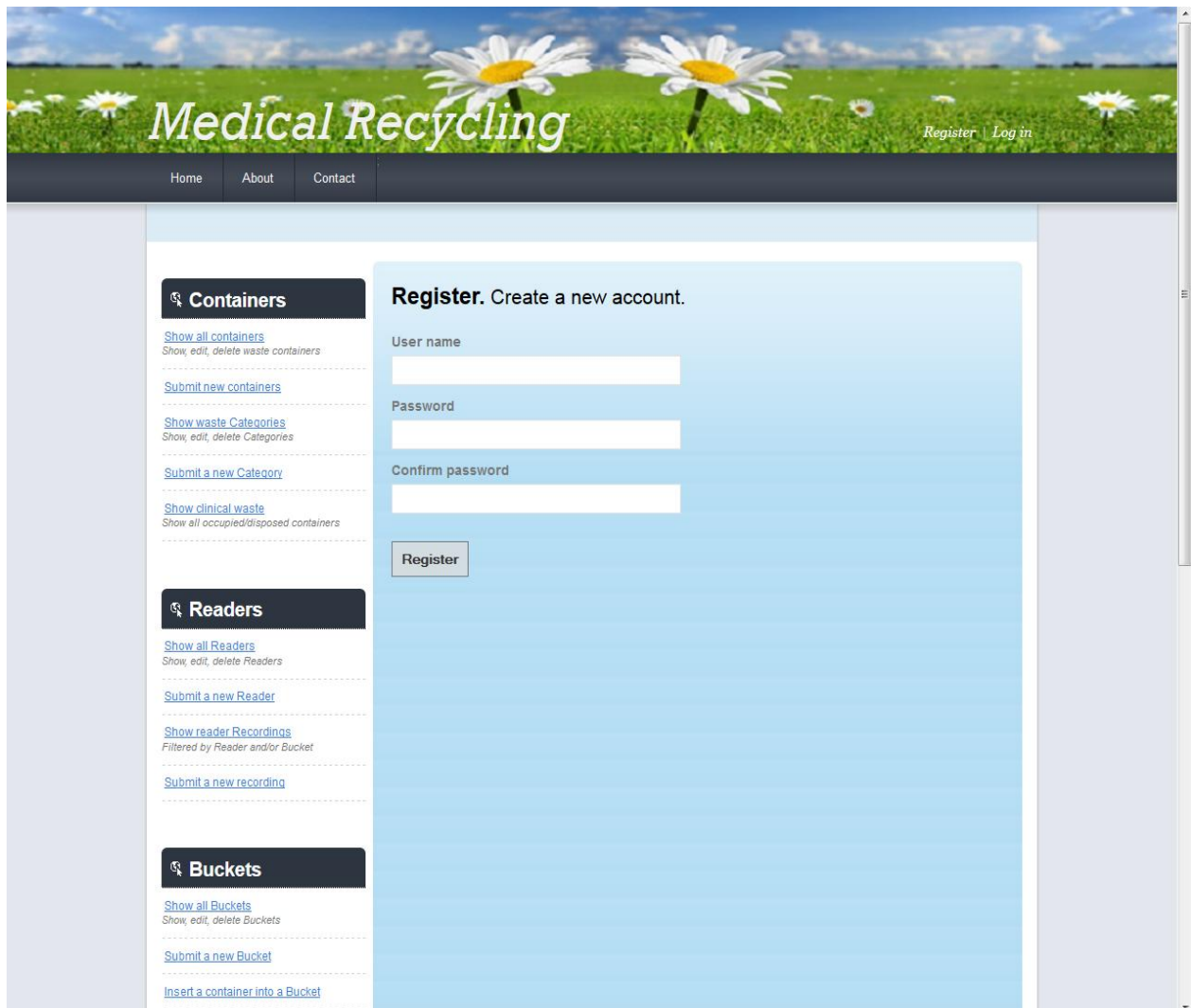
Τέλος, υπάρχει περίπτωση ο χρήστης να εισάγει λανθασμένα στοιχεία εισόδου, και ενημερώνεται κατάλληλα:



Εικόνα 5.11. Αποτυχημένο log in.

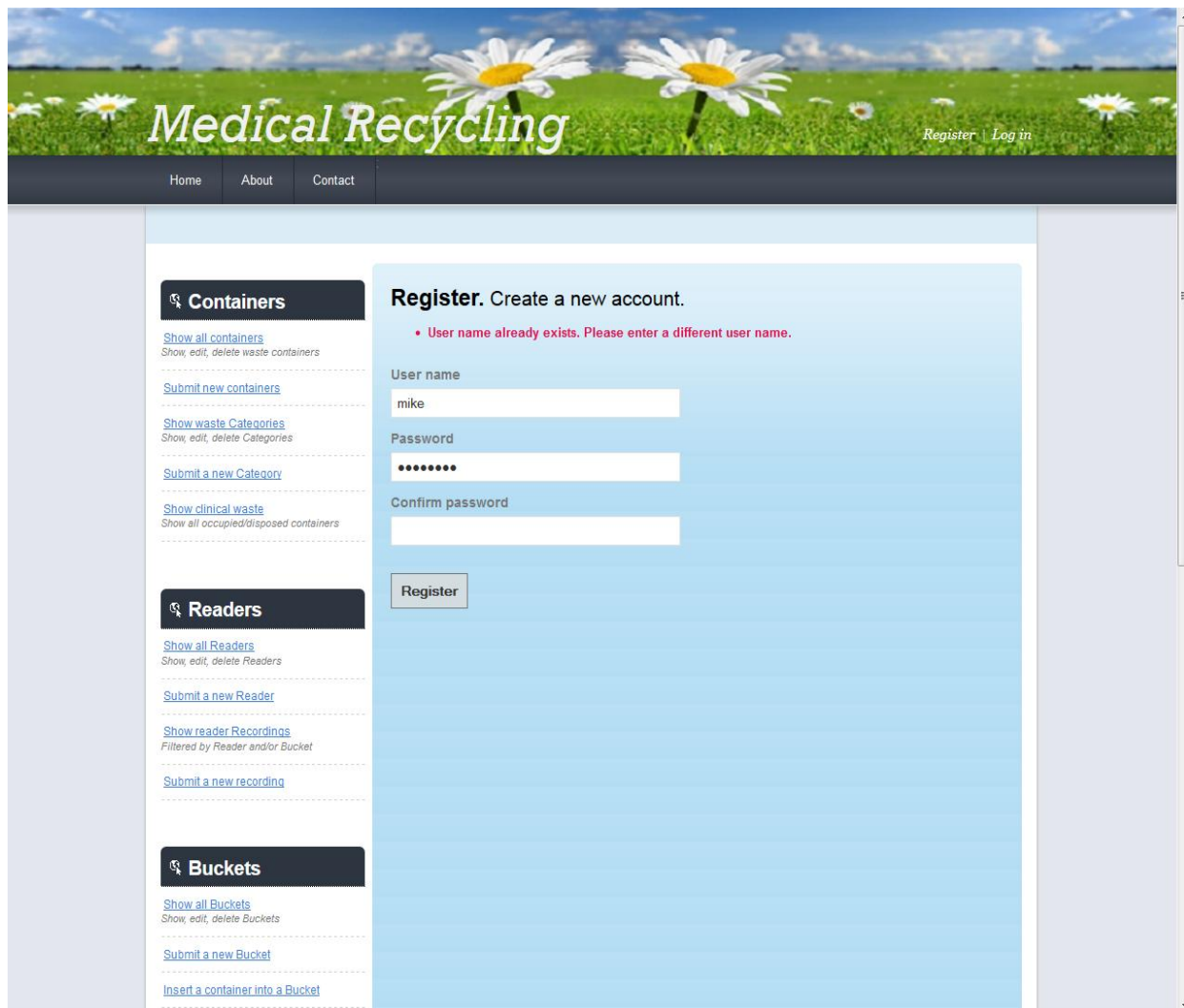
5.2 Εγγραφή (Registration) χρήστη

Στο σενάριο αυτό ο χρήστης επιχειρεί να δημιουργήσει έναν λογαριασμό για την ιστοσελίδα μας.

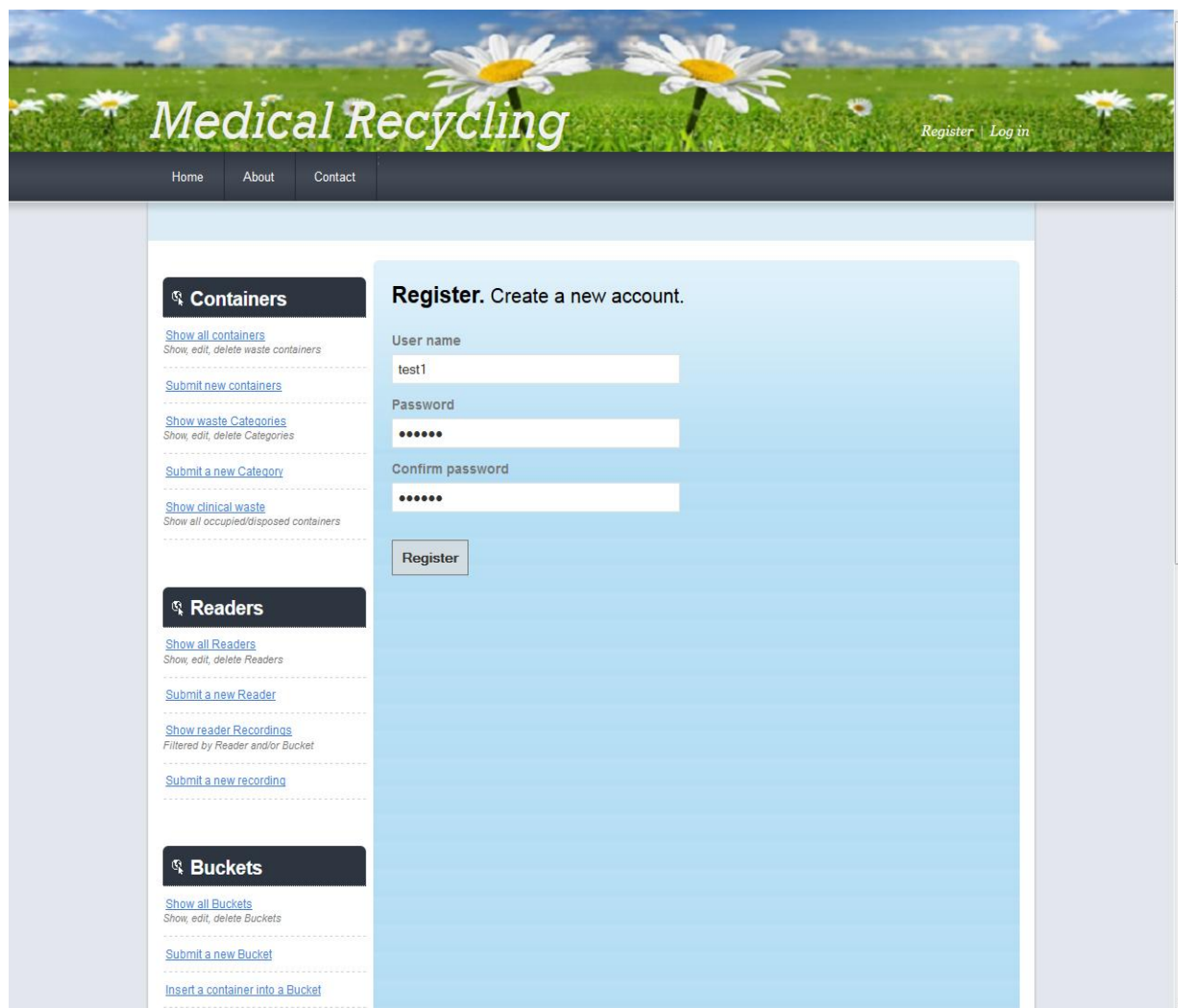


Εικόνα 5.12. Σελίδα εγγραφής (registration) χρήστη.

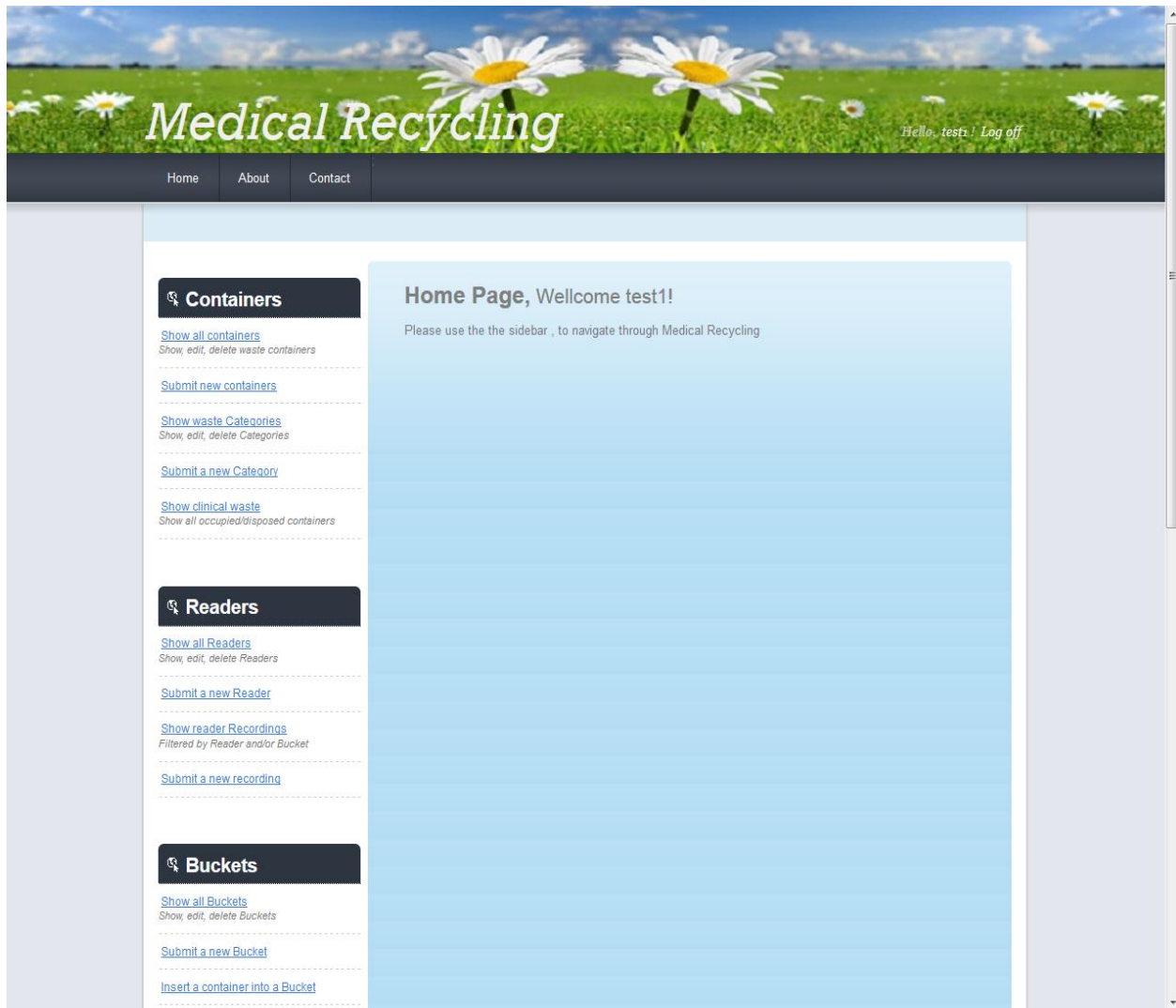
Ο χρήστης καλείται να δώσει ένα όνομα χρήστη (user name) και τον κωδικό του (password), επαληθευόντάς τον. Προφανώς ελέγχεται αν το όνομα χρήστη που εισάγεται είναι κατειλημμένο από κάποιον άλλον, και αν συμβαίνει αυτό, εμφανίζεται κατάλληλο ενημερωτικό μήνυμα.



Εικόνα 5.13. Αποτυχημένη εγγραφή.

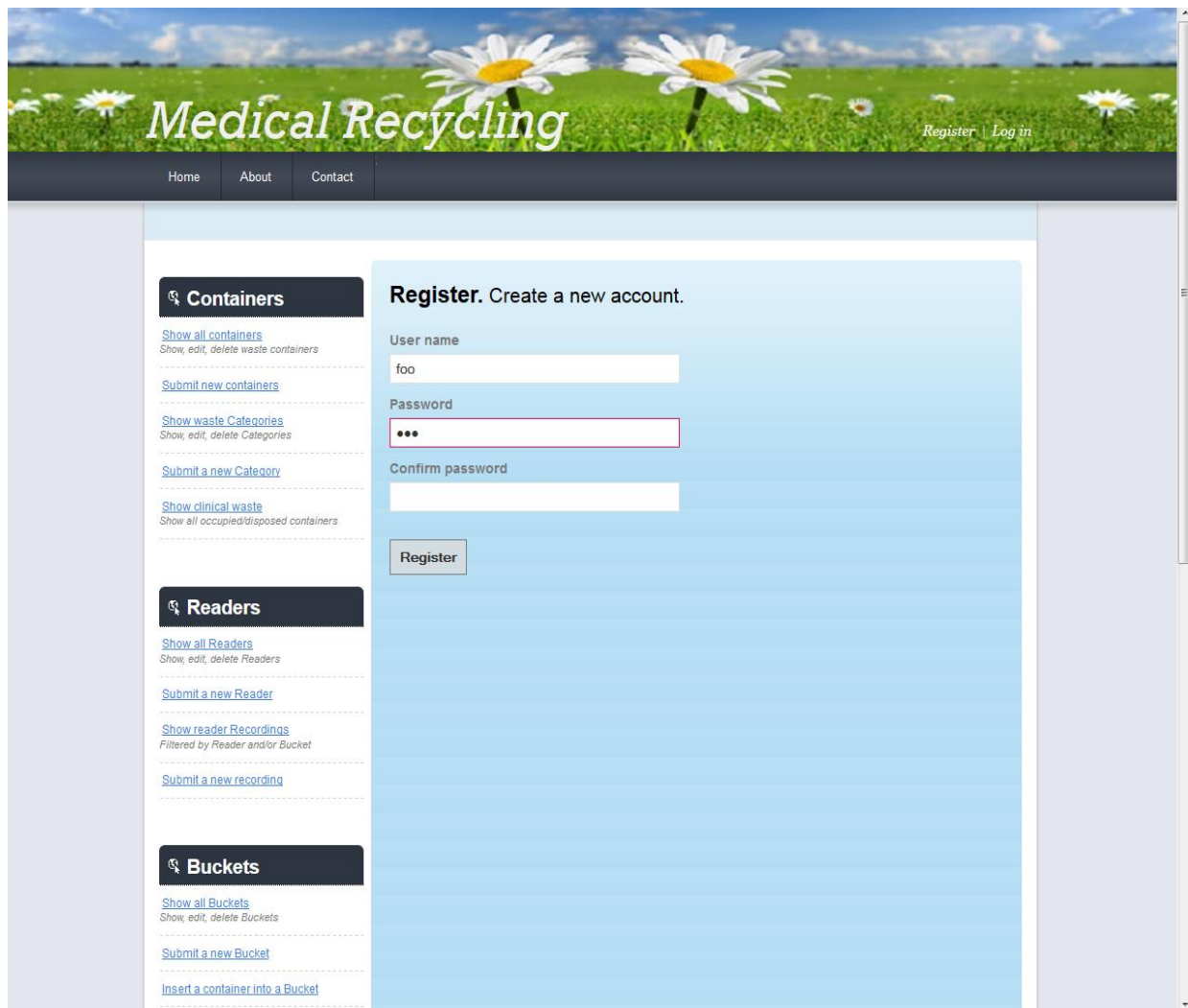


Εικόνα 5.14. Δεύτερη απόπειρα εγγραφής.



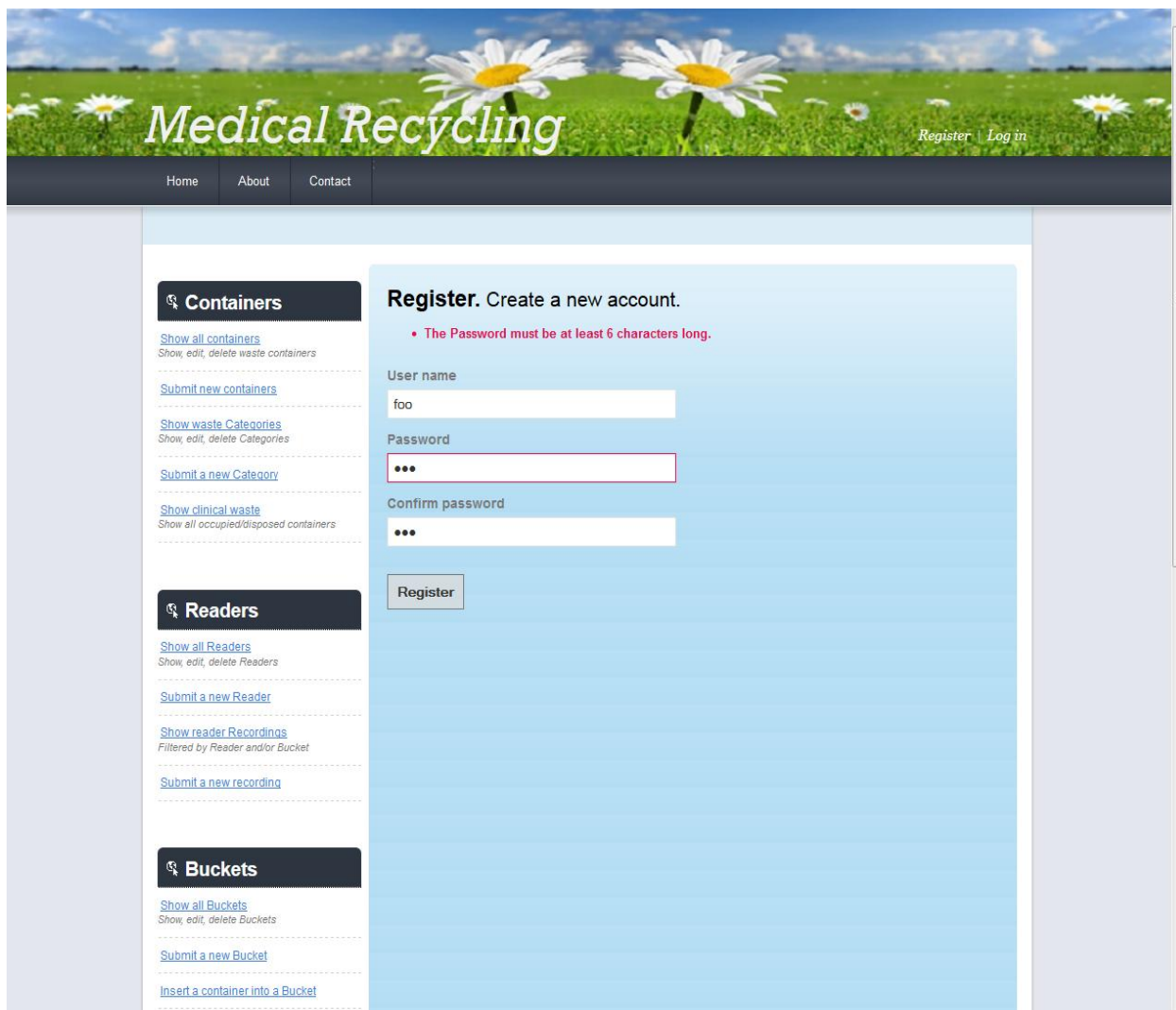
Εικόνα 5.15. Ολοκλήρωση Εγγραφής.

Ακόμη ελέγχεται αν ο κωδικός που έδωσε ο χρήστης είναι μεγαλύτερος από 6 χαρακτήρες. Αυτός ο περιορισμός έχει επιβληθεί για λόγους ασφάλειας. Αν ο κωδικός είναι μικρότερος, και ο χρήστης απομακρυνθεί από το συγκεκριμένο πεδίο (κάνει tab – off), με χρήση της γλώσσας JavaScript, το συγκεκριμένο πεδίο μαρκάρεται με κόκκινο χρώμα, θέλοντας να δείξει πως είναι λανθασμένο.



Εικόνα 5.16. Μικρός κωδικός πρόσβασης.

Αν ο χρήστης συνεχίσει και επιβεβαιώσει τον ίδιο (μικρό) κωδικό, και πιάσει το κουμπί Register, τότε πάλι με χρήση JavaScript, αποφεύγουμε να στείλουμε τα λανθασμένα δεδομένα στον Server, και εμφανίζεται κατάλληλο ενημερωτικό μήνυμα που τον προτρέπει να χρησιμοποιήσει έναν κωδικό μεγαλύτερο των 6 χαρακτήρων.



Εικόνα 5.17. Μικρός κωδικός πρόσβασης – επιβεβαίωση.

5.3 Διαχείριση των Περιεκτών (Containers)

5.3.1 Επισκόπηση, τροποποίηση και διαγραφή Περιεκτών

Σε αυτό το σενάριο ο χρήστης (ο οποίος έχει ταυτοποιηθεί) επιθυμεί να δει τους καταχωρημένους περιέκτες στην νοσοκομειακή μονάδα. Ακολουθώντας το Link, που βρίσκεται στο αντικείμενο Containers του sidebar, *Show all containers*, ο χρήστης βλέπει και μπορεί να επεξεργάζεται τα καταχωρημένα στοιχεία των περιεκτών, όπως τον τύπο τους (Plastic Bag – Hospital Box), το RFIDtag τους, το μέγεθός τους, την κατηγορία τους, και το χρώμα τους. Θεωρήθηκε ότι δεν έχει νόημα η επεξεργασία πληροφοριών όπως η ημερομηνία παραλαβής

(καταχώρησης) των περιεκτών, και το αν είναι άδειοι ή όχι, αφού οι ενέργειες αυτές γίνονται αυτόματα, και δεν πρέπει να τροποποιηθούν χειροκίνητα.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

Show all containers
Show, edit, delete waste containers

Submit new containers

Show waste Categories
Show, edit, delete Categories

Submit a new Category

Show clinical waste
Show all occupied/disposed containers

Readers

Show all Readers
Show, edit, delete Readers

Submit a new Reader

Show reader Recordings
Filtered by Reader and/or Bucket

Submit a new recording

Buckets

Show all Buckets
Show, edit, delete Buckets

Submit a new Bucket

Insert a container into a Bucket

Empty a Bucket in a Storeroom
Empty an occupied bucket

Containers

Group by: RFIDtag | Category | Date Arrived

Type	RFIDtag	Date Arrived	Date Expires	Size	Empty	Category	Color	
Plastic Bag	d860f83c d2cf-460 ... 327f597e...	3/6/2013	11/11/2015	5	<input type="checkbox"/>	IA-AX	Red	Edit Delete
Plastic Bag	aa4d796 ... a070-522...	3/6/2013	11/11/2015	5	<input type="checkbox"/>	IA-AX	Red	Edit Delete
Plastic Bag	9ac24eda 6b3c-49ef ... 9056-5cfe...	3/6/2013	1/1/2014	6	<input checked="" type="checkbox"/>	EIA-TX	Red	Edit Delete
Plastic Bag	0e9f2aa6-... b0033a6e...	3/6/2013	1/1/2014	4	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	1ab1f7bb 8860-405...	3/18/2013	12/22/2014	7	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	0a3480f8-... 41eb-bd5f ... 6386a430...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	b18ae617-... a097-f376...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	48ba85e ba73-4ed ... a6ec-2f29...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	5a61aae fe96-461 ... b620-81d...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	fc63937a 23e5-42c ... a67c-3de...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	ea44a69 ab4e-44 ... 98c1-db7...	3/18/2013	12/12/2014	11	<input type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	ee1cb7df aca5-48a ... be2e-40d...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	7aaa6431 e94b-453... 521126fae...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	f9e691ac- 3732-4fef- 917c-cd98...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	ce763d9 dd9b-47c...	4/4/2013	5/12/2015	11	<input checked="" type="checkbox"/>	EIA-MTX	Yellow	Edit Delete

Εικόνα 5.18. Σελίδα – Περιέκτες (Containers).

Βλέπουμε ότι στο τέλος κάθε γραμμής του πίνακα υπάρχει η επιλογή για διαγραφή ή τροποποίηση των καταχωρημένων στοιχείων για κάθε περιέκτη. Μετά από κάποια από αυτές τις δύο ενέργειες, καταχωρούνται οι σχετικές αλλαγές στην βάση δεδομένων. Προφανώς, κατά την τροποποίηση κάποιων από τις καταχωρημένες τιμές για τα πεδία των περιεκτών γίνεται έλεγχος για το αν τα εισαχθέντα δεδομένα είναι έγκυρα. Ο έλεγχος αυτός είναι όμοιος με αυτόν κατά την διαδικασία ενός νέου περιέκτη, γι αυτό θα παρουσιαστεί παρακάτω.

Επειδή οι περιέκτες συνήθως καταχωρούνται ομαδικά (δεν γίνονται παραγγελίες πχ ενός κίτρινου, ενός κόκκινου, και ενός μαύρου περιέκτη – αλλά ομάδες από αυτούς πχ 100 κόκκινοι, 100 κίτρινοι, 100 μαύροι), υπάρχει επιλογή ομαδοποίησης (Group By) των περιεκτών, ανά κατηγορία ή ημερομηνία άφιξης. Οι σχετικές επιλογές βρίσκονται ακριβώς επάνω από τον πίνακα στον οποίο εμφανίζονται οι περιέκτες.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

[Show all containers](#)
Show, edit, delete waste containers

[Submit new containers](#)

[Show waste Categories](#)
Show, edit, delete Categories

[Submit a new Category](#)

[Show clinical waste](#)
Show all occupied/disposed containers

Group by: [RFID tag](#) | [Category](#) | [Date Arrived](#)

Count	Type	Color	Size	Category	DateArrived	DateExpires
20	Hospital box	-	5	EIA-MTX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM
100	Plastic Bag	Black	3	IA-AX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM
1	Plastic Bag	Red	4	AIA	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
9	Plastic Bag	Red	11	AIA	3/18/2013 12:00:00 AM	12/12/2014 12:00:00 AM
1	Plastic Bag	Red	7	AIA	3/18/2013 12:00:00 AM	12/22/2014 12:00:00 AM
1	Plastic Bag	Red	4	AIA	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
1	Plastic Bag	Red	2	EIA-MTX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
1	Plastic Bag	Red	10	EIA-MTX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
2	Plastic Bag	Red	5	EIA-MX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
3	Plastic Bag	Red	5	EIA-MX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
1	Plastic Bag	Red	6	EIA-TX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
1	Plastic Bag	Red	6	EIA-TX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
2	Plastic Bag	Red	3	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
3	Plastic Bag	Red	5	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
5	Plastic Bag	Yellow	4	AIA	4/11/2013 12:00:00 AM	12/12/2015 12:00:00 AM
11	Plastic Bag	Yellow	11	EIA-MTX	4/4/2013 12:00:00 AM	5/12/2015 12:00:00 AM
2	Plastic Bag	Yellow	3	IA-AX	5/14/2013 12:00:00 AM	4/4/2014 12:00:00 AM

Readers

[Show all Readers](#)
Show, edit, delete Readers

[Submit a new Reader](#)

[Show reader Recordings](#)
Filtered by Reader and/or Bucket

[Submit a new recording](#)

Buckets

[Show all Buckets](#)
Show, edit, delete Buckets

[Submit a new Bucket](#)

[Insert a container into a Bucket](#)

Εικόνα 5.19. Περιέκτες – grouped by Category.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

[Show all containers](#)
Show, edit, delete waste containers

[Submit new containers](#)

[Show waste Categories](#)
Show, edit, delete Categories

[Submit a new Category](#)

[Show clinical waste](#)
Show all occupied/disposed containers

Readers

[Show all Readers](#)
Show, edit, delete Readers

[Submit a new Reader](#)

[Show reader Recordings](#)
Filtered by Reader and/or Bucket

[Submit a new recording](#)

Buckets

[Show all Buckets](#)
Show, edit, delete Buckets

[Submit a new Bucket](#)

[Insert a container into a Bucket](#)

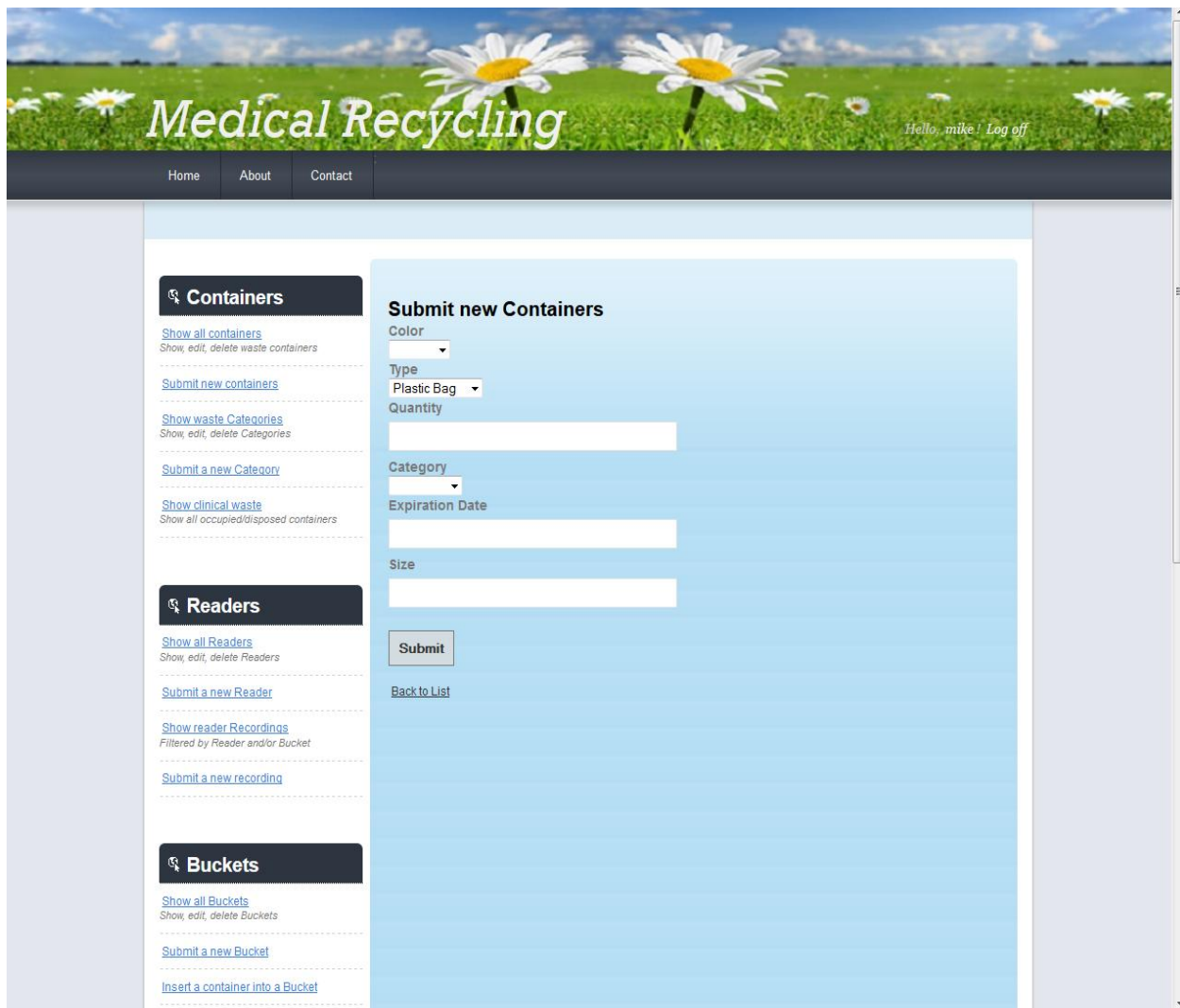
Group by: [RFID tag](#) | [Category](#) | [DateArrived](#)

Count	Type	CATEGORY	DateArrived	DateExpires	Size	Color
20	Hospital box	EIA-MTX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM	5	-
100	Plastic Bag	IA-AX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM	3	Black
1	Plastic Bag	AIA	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	4	Red
1	Plastic Bag	AIA	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	4	Red
1	Plastic Bag	EIA-MTX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	2	Red
1	Plastic Bag	EIA-MTX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	10	Red
2	Plastic Bag	EIA-MX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	5	Red
3	Plastic Bag	EIA-MX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	5	Red
1	Plastic Bag	EIA-TX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	6	Red
1	Plastic Bag	EIA-TX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	6	Red
2	Plastic Bag	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	3	Red
3	Plastic Bag	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	5	Red
1	Plastic Bag	AIA	3/18/2013 12:00:00 AM	12/22/2014 12:00:00 AM	7	Red
9	Plastic Bag	AIA	3/18/2013 12:00:00 AM	12/12/2014 12:00:00 AM	11	Red
11	Plastic Bag	EIA-MTX	4/4/2013 12:00:00 AM	5/12/2015 12:00:00 AM	11	Yellow
5	Plastic Bag	AIA	4/11/2013 12:00:00 AM	12/12/2015 12:00:00 AM	4	Yellow
2	Plastic Bag	IA-AX	5/14/2013 12:00:00 AM	4/4/2014 12:00:00 AM	3	Yellow

Εικόνα 5.20. Περιέκτες – Grouped By Date Arrived.

5.3.2 Καταχώρηση νέων περιεκτών

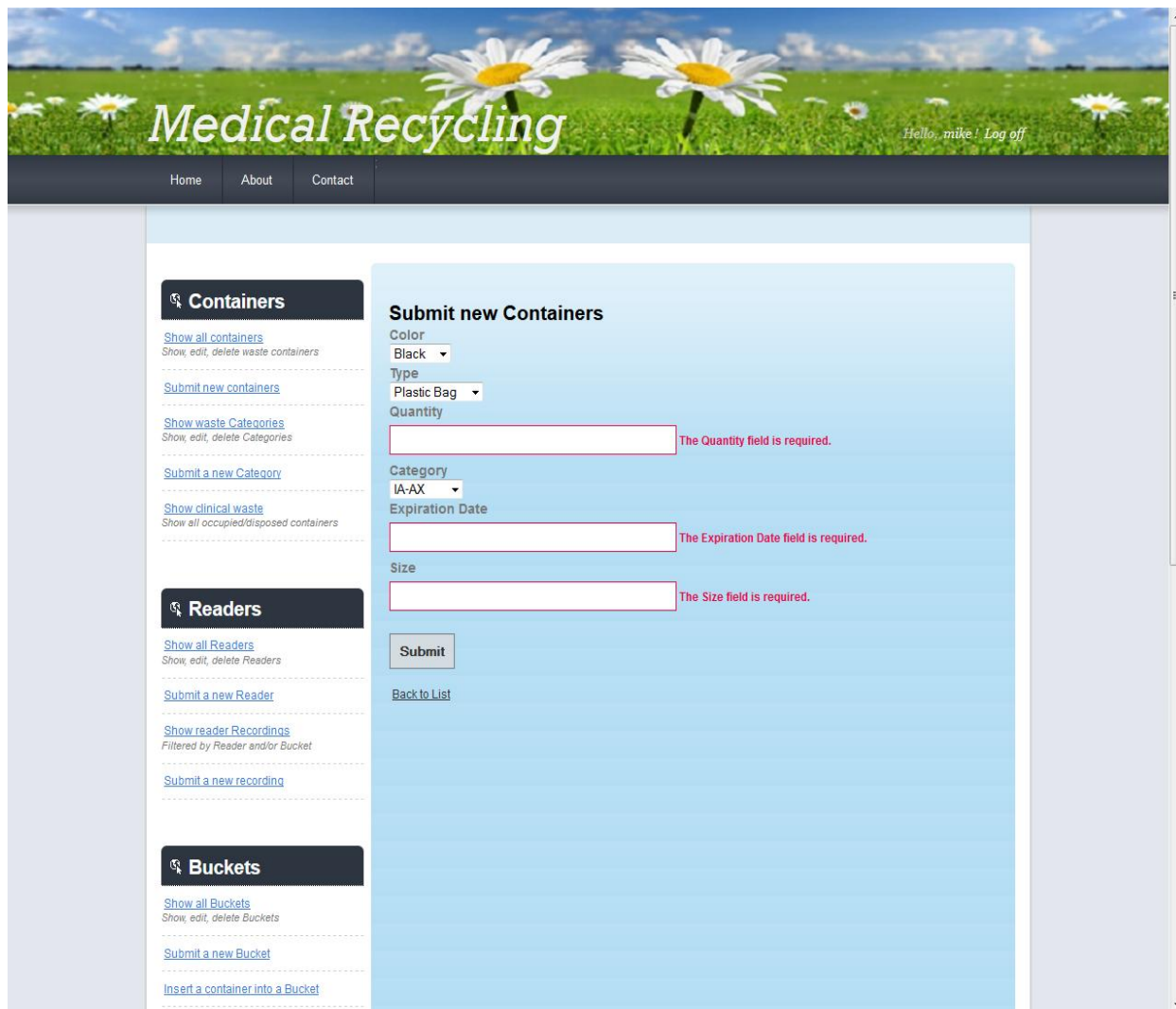
Στο σενάριο αυτό ο χρήστης επιχειρεί να καταχωρήσει νέους περιέκτες στο σύστημα. Καλείται λοιπόν να συμπληρώσει τον τύπο τους, το χρώμα τους (απαραίτητο πεδίο αν ο Περιέκτης είναι τύπου Plastic Bag), την ποσότητα, την κατηγορία ιατρικών αποβλήτων που μπορούν να περιέχουν, την ημερομηνία λήξης τους και το μέγεθός τους. Θεωρήθηκε ότι η ημερομηνία παραλαβής των Περιεκτών ταυτίζεται με την ημερομηνία καταχώρησής τους στο σύστημα, και ότι το RFID tag δίδεται αυτόματα, και συνεπώς ο χρήστης δεν χρειάζεται (και δεν πρέπει) να συμπληρώσει αυτά τα πεδία.



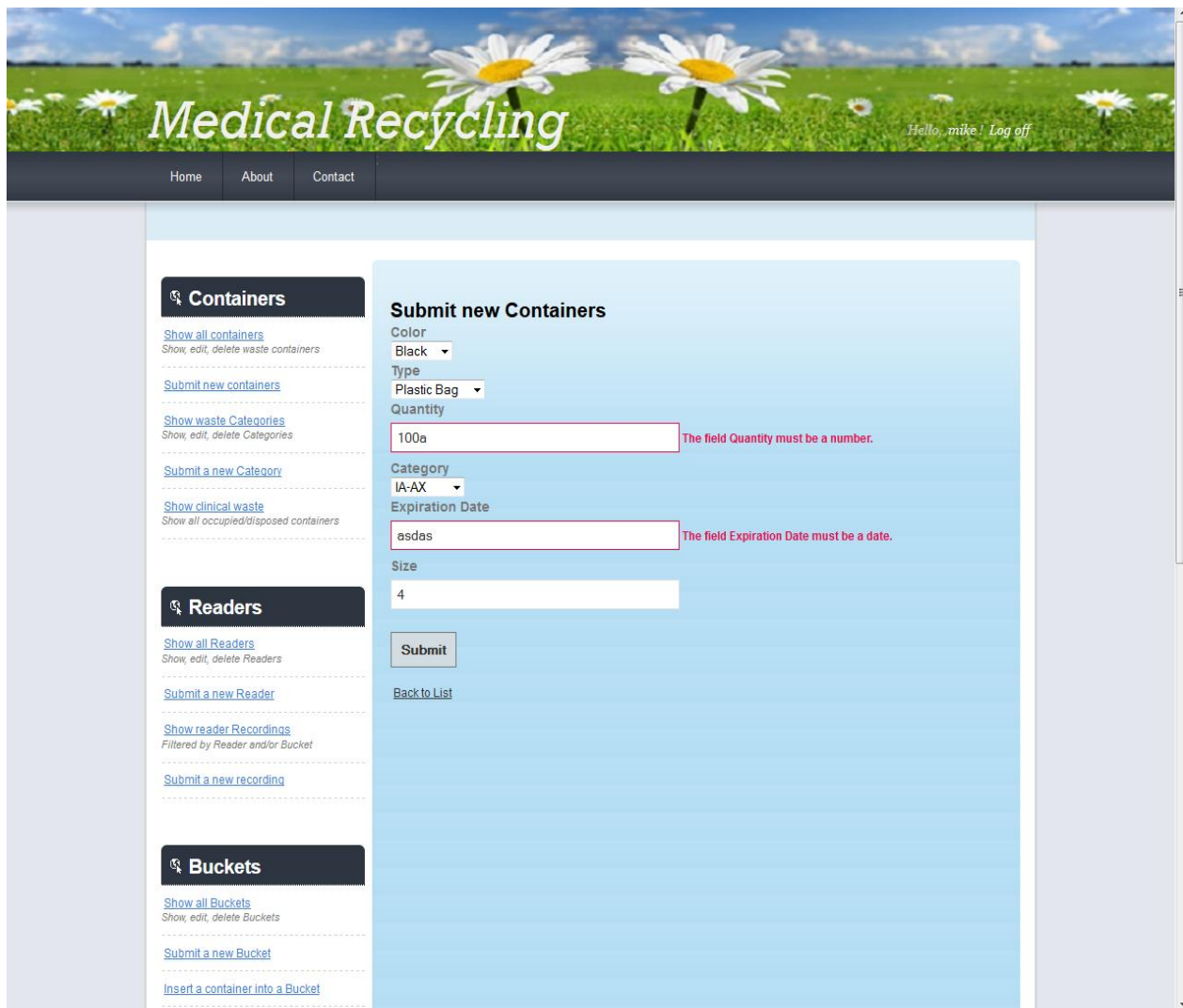
Εικόνα 5.21. Καταχώρηση νέων περιεκτών.

Η αποφυγή της εισαγωγής λανθασμένων στοιχείων στην βάση δεδομένων του συστήματος είναι κομβικής σημασίας για την αξιοπιστία και την εγγυημένη λειτουργία του, και γι αυτόν τον σκοπό εφαρμόζεται έλεγχος, και εμφανίζονται κατάλληλα ενημερωτικά μηνύματα στην περίπτωση λανθασμένων στοιχείων. Έχει γίνει προσπάθεια διευκόλυνσης – καθοδήγησης του χρήστη μέσω Drop Down Lists , Select Boxes και αναλυτικά μηνύματα λανθασμένων δεδομένων.

Αν ο χρήστης επιχειρήσει να καταχωρήσει νέους περιέκτες χωρίς να συμπληρώσει όλα τα απαραίτητα στοιχεία γι αυτούς ενημερώνεται κατάλληλα. Όλοι οι έλεγχοι των πεδίων που θα ακολουθήσουν γίνονται με χρήση JavaScript, έτσι ώστε να αποφεύγεται η μεταφορά λανθασμένων στοιχείων στον Server. Η λογική αυτή έχει εφαρμοσθεί σε όλες της φόρμες του ιστότοπου που ζητείται είσοδος από τον χρήστη.



Εικόνα 5.22. Περιέκτες - Ελληνική στοιχεία.



Εικόνα 5.23. Περιέκτες - Λανθασμένα στοιχεία

Σε περίπτωση που τα δεδομένα που εισήγαγε ο χρήστης είναι σωστά, δημιουργούνται οι αντίστοιχες εγγραφές για τους νέους περιέκτες στην βάση δεδομένων. Δημιουργούνται αυτόματα το RFID tag και καταχωρείται η ημερομηνία παραλαβής ως η ημερομηνία καταχώρησης, και προφανώς καταχωρείται by default ότι ο κάθε περιέκτης είναι αρχικά άδειος.

5.4 Διαχείριση των Κατηγοριών (Categories)

5.4.1 Επισκόπηση και διαχείριση των κατηγοριών ιατρικών αποβλήτων

Κατ' απόλυτη αναλογία με το προηγούμενο σενάριο ο χρήστης επιθυμεί να δει τις καταχωρημένες κατηγορίες ιατρικών αποβλήτων και όλες τις απαραίτητες πληροφορίες που σχετίζονται με αυτές (έχουν παρουσιαστεί στο προηγούμενο κεφάλαιο), και να τροποποιήσει κάποια κατηγορία εφ' όσον το επιθυμεί, ακολουθώντας το link *Show all Categories* που βρίσκεται στο αντικείμενο Containers του sidebar. Να σημειωθεί σε αυτό το σημείο ότι για λόγους πληρότητας των δεδομένων (η κατηγορία ιατρικών αποβλήτων είναι εξωτερικό κλειδί στον πίνακα Περιέκτες (Containers)) δεν επιτρέπεται η διαγραφή κάποιας κατηγορίας ιατρικών αποβλήτων.

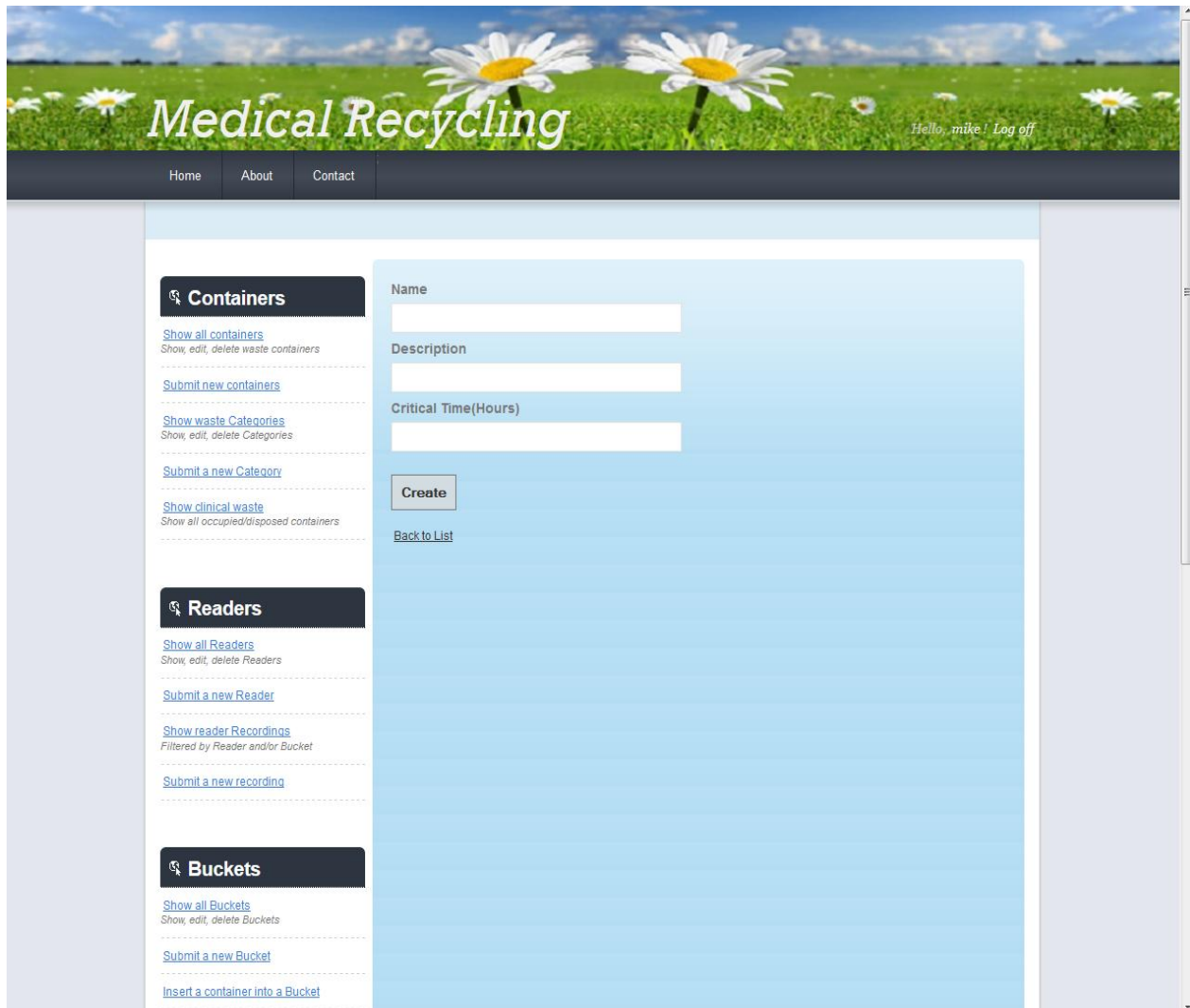
The screenshot shows the 'Medical Recycling' website interface. The header features the site name and a user greeting 'Hello, mike! Log off'. A navigation menu includes 'Home', 'About', and 'Contact'. The main content area is organized into three sidebar sections: 'Containers', 'Readers', and 'Buckets'. The 'Categories' section is the focus, displaying a table of medical waste categories.

Name	Description	Critical Time(Hours)	
IA-AX	Clinical waste - Urban	10.00	Edit
EIA-MX	Hazardous Clinical Waste - Purerly Infectious	5.00	Edit
EIA-MTX	Hazardous Clinical Waste - Purerly Toxic, Infectious	5.00	Edit
EIA-TX	Hazardous Clinical Waste - Purerly Toxic	3.00	Edit
AIA	Other Clinical Waste	12.00	Edit

Εικόνα 5.23. Κατηγορίες ιατρικών αποβλήτων.

5.4.2 Καταχώρηση νέας κατηγορίας Ιατρικών Αποβλήτων

Ο χρήστης μπορεί να καταχωρήσει κάποια καινούρια κατηγορία ιατρικών αποβλήτων συμπληρώνοντας όλα τα απαραίτητα στοιχεία σχετικά με αυτήν. Και σε αυτήν την περίπτωση εφαρμόζεται έλεγχος των πεδίων που συμπληρώνει ο χρήστης.



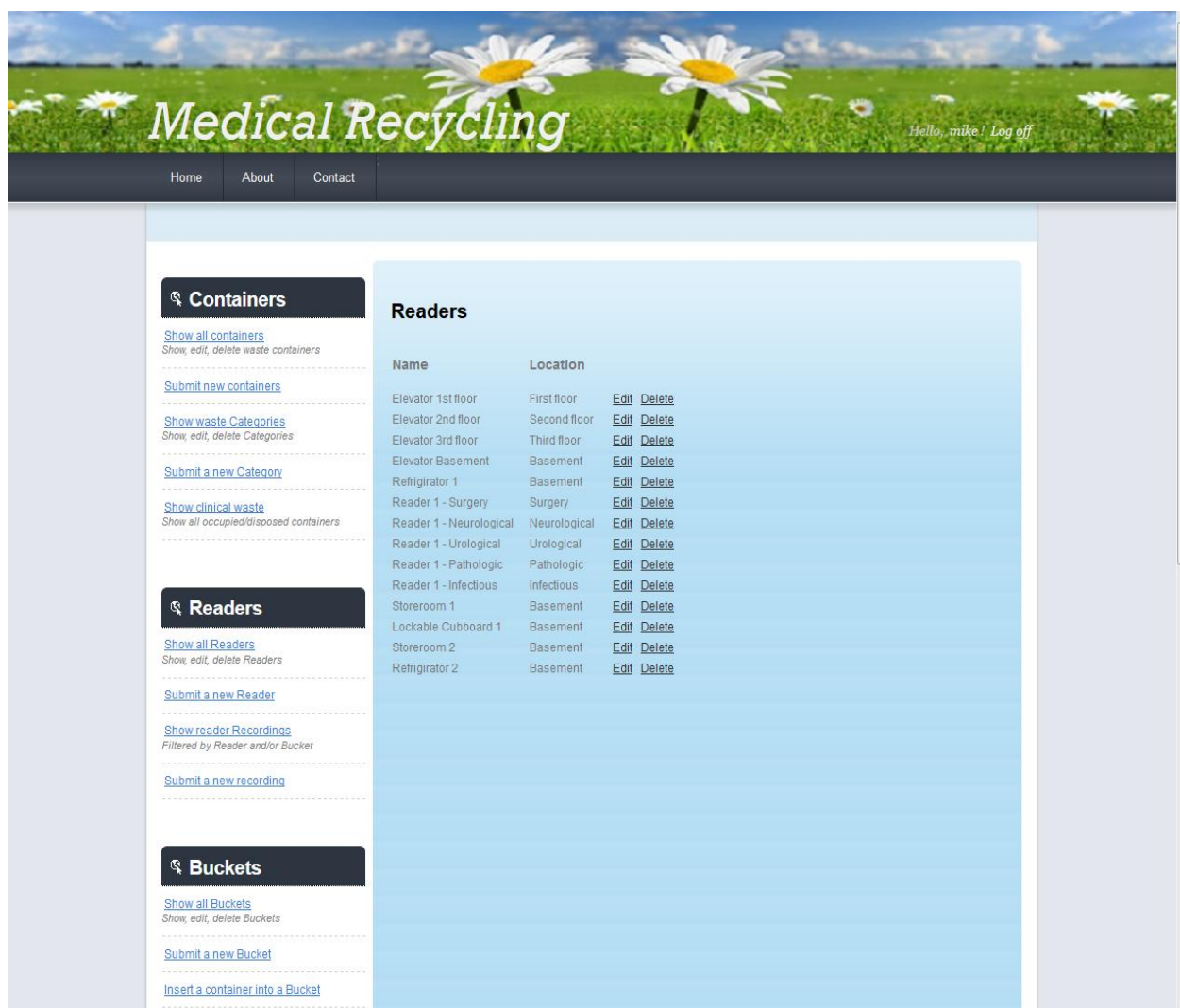
The screenshot shows the 'Medical Recycling' website interface. At the top, there is a banner with a field of daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into three sections: 'Containers', 'Readers', and 'Buckets'. Each section has a 'Show all' link and a 'Submit a new' link. The 'Containers' section is currently active, showing a form with the following fields: 'Name', 'Description', and 'Critical Time(Hours)'. There is a 'Create' button and a 'Back to List' link below the form.

Εικόνα 5.23. Δημιουργία νέας κατηγορίας Ι.Α.

5.5 Διαχείριση των Αναγνωστών (Readers)

5.5.1 Επισκόπηση, τροποποίηση και διαγραφή Αναγνώστων

Σε αυτό το σενάριο ο χρήστης επιχειρεί να δει όλους τους καταχωρημένους Αναγνώστες της νοσοκομειακής μονάδας, και όλων των απαραίτητων στοιχείων που σχετίζονται με αυτούς και να τροποποιήσει κάποιο στοιχείο από κάποιον αναγνώστη που έχει ήδη εισαχθεί ή να διαγράψει κάποιον αναγνώστη. Για να χρησιμοποιήσει αυτήν την λειτουργία ακολουθεί το link *Show all Readers* που βρίσκεται στο αντικείμενο Readers του sidebar.



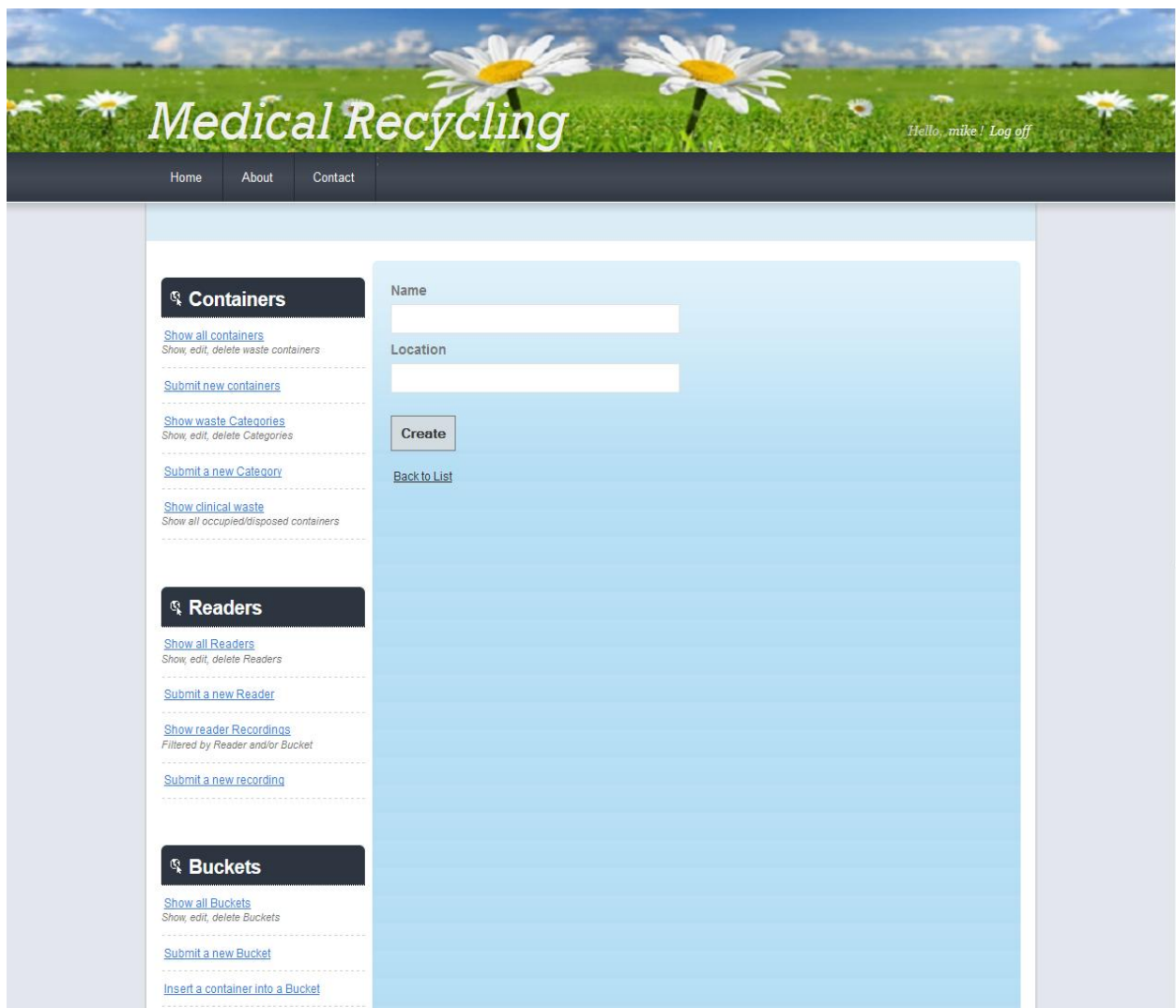
The screenshot shows the 'Medical Recycling' website interface. The header features a banner with daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into three sections: 'Containers', 'Readers', and 'Buckets'. The 'Readers' section is active and displays a table of readers with columns for Name, Location, and actions (Edit, Delete).

Name	Location	
Elevator 1st floor	First floor	Edit Delete
Elevator 2nd floor	Second floor	Edit Delete
Elevator 3rd floor	Third floor	Edit Delete
Elevator Basement	Basement	Edit Delete
Refrigerator 1	Basement	Edit Delete
Reader 1 - Surgery	Surgery	Edit Delete
Reader 1 - Neurological	Neurological	Edit Delete
Reader 1 - Urological	Urological	Edit Delete
Reader 1 - Pathologic	Pathologic	Edit Delete
Reader 1 - Infectious	Infectious	Edit Delete
Storeroom 1	Basement	Edit Delete
Lockable Cubboard 1	Basement	Edit Delete
Storeroom 2	Basement	Edit Delete
Refrigerator 2	Basement	Edit Delete

Εικόνα 5.25. Αναγνώστες (Readers).

5.5.2 Καταχώρηση νέου Αναγνώστη

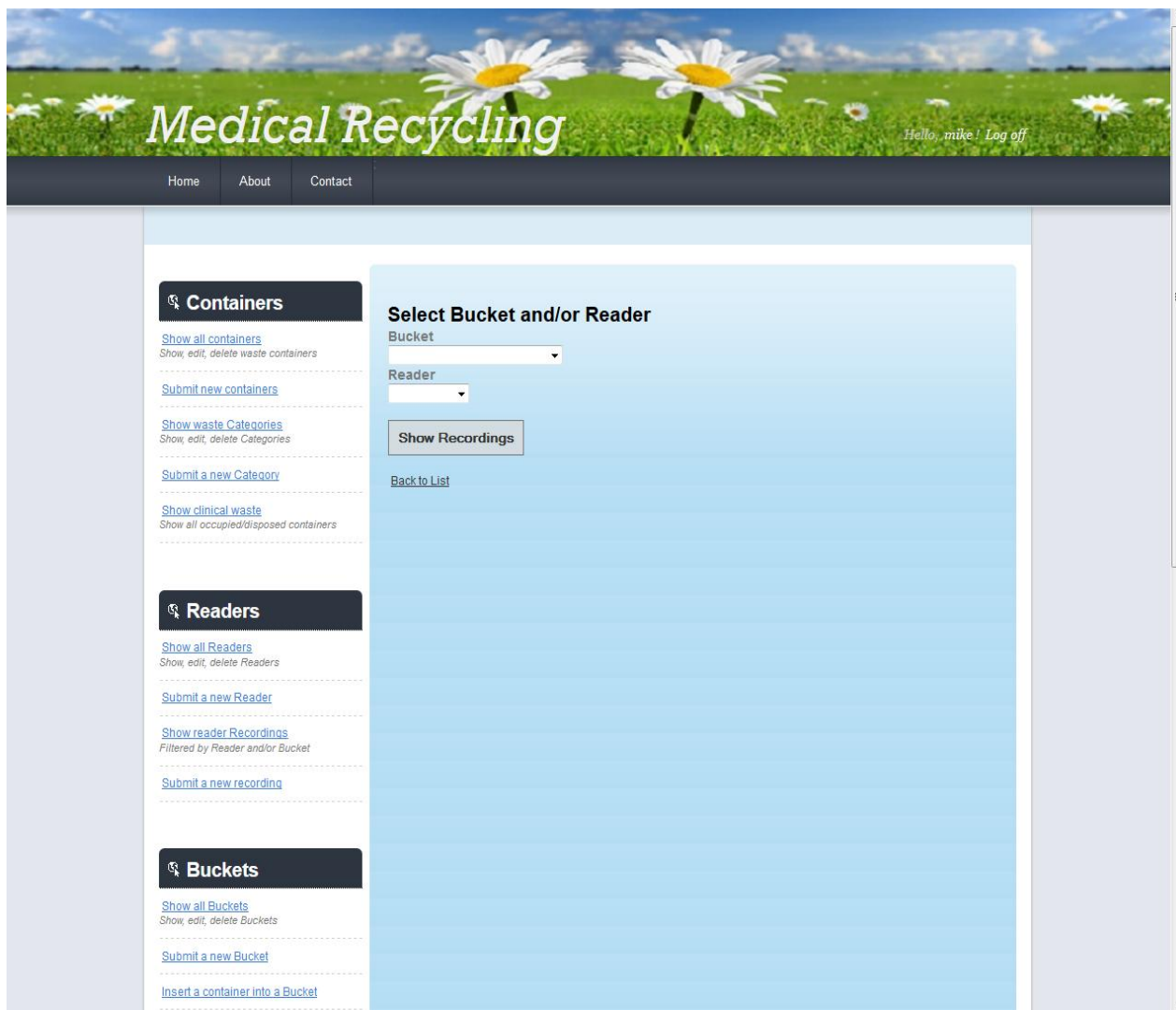
Σε αυτό το σενάριο ο χρήστης επιχειρεί να καταχωρήσει έναν νέο αναγνώστη στο σύστημα, συμπληρώνοντας όλα τα απαραίτητα στοιχεία που σχετίζονται με αυτόν, όπως αυτά έχουν παρουσιαστεί, ακολουθώντας το link *Submit a new Reader* που βρίσκεται στο αντικείμενο Readers του Sidebar. Κατ' απόλυτη αναλογία με τα προηγούμενα σενάρια, εφαρμόζεται έλεγχος των δεδομένων που εισάγει ο χρήστης.



Εικόνα 5.27. Καταχώρηση νέου Αναγνώστη.

5.5.3 Εμφάνιση των καταγραφών (recordings) του Αναγνώστη

Σε αυτό το σενάριο ο χρήστης επιθυμεί να δει τις καταγραφές των Αναγνώστών. Για να χρησιμοποιήσει αυτήν την λειτουργία ακολουθεί το link *Show Reader Recordings* που βρίσκεται στο αντικείμενο Readers του sidebar.



Εικόνα 5.28. Επιλογή Αναγνώστη ή Συλλέκτη.

Η σελίδα που εμφανίζεται στον χρήστη τον προτρέπει να επιλέξει κάποιον συλλέκτη η / και κάποιον αναγνώστη. Αν επιλεγεί μόνο κάποιος συλλέκτης, εμφανίζονται οι κινήσεις του συλλέκτη μέσα στην νοσοκομειακή μονάδα. Εμφανίζονται δηλαδή όλες οι καταγραφές του συγκεκριμένου συλλέκτη από όλους τους αναγνώστες. Αν επιλεγεί μόνο κάποιος αναγνώστης τότε εμφανίζονται όλες οι καταγραφές του συγκεκριμένου αναγνώστη. Αν επιλεγεί και συλλέκτης και αναγνώστης τότε εμφανίζονται όλες οι καταγραφές του συγκεκριμένου συλλέκτη από τον συγκεκριμένο αναγνώστη. Να σημειωθεί σε αυτό το σημείο ότι στην Drop Down List που εμφανίζεται στον χρήστη για να επιλέξει Συλλέκτη και Αναγνώστη, εμφανίζονται μόνο οι συλλέκτες που έχουν καταγραφεί από κάποιον αναγνώστη, και αντίστοιχα οι αναγνώστες που έχουν κάνει κάποια καταγραφή. Δεν θα είχε νόημα για παράδειγμα να εμφανίσουμε τις καταγραφές κάποιου αναγνώστη ο οποίος δεν έχει καταγράψει τίποτα.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

[Show all containers](#)
Show, edit, delete waste containers

[Submit new containers](#)

[Show waste Categories](#)
Show, edit, delete Categories

[Submit a new Category](#)

[Show clinical waste](#)
Show all occupied/disposed containers

Readers

[Show all Readers](#)
Show, edit, delete Readers

[Submit a new Reader](#)

[Show reader Recordings](#)
Filtered by Reader and/or Bucket

[Submit a new recording](#)

Buckets

[Show all Buckets](#)
Show, edit, delete Buckets

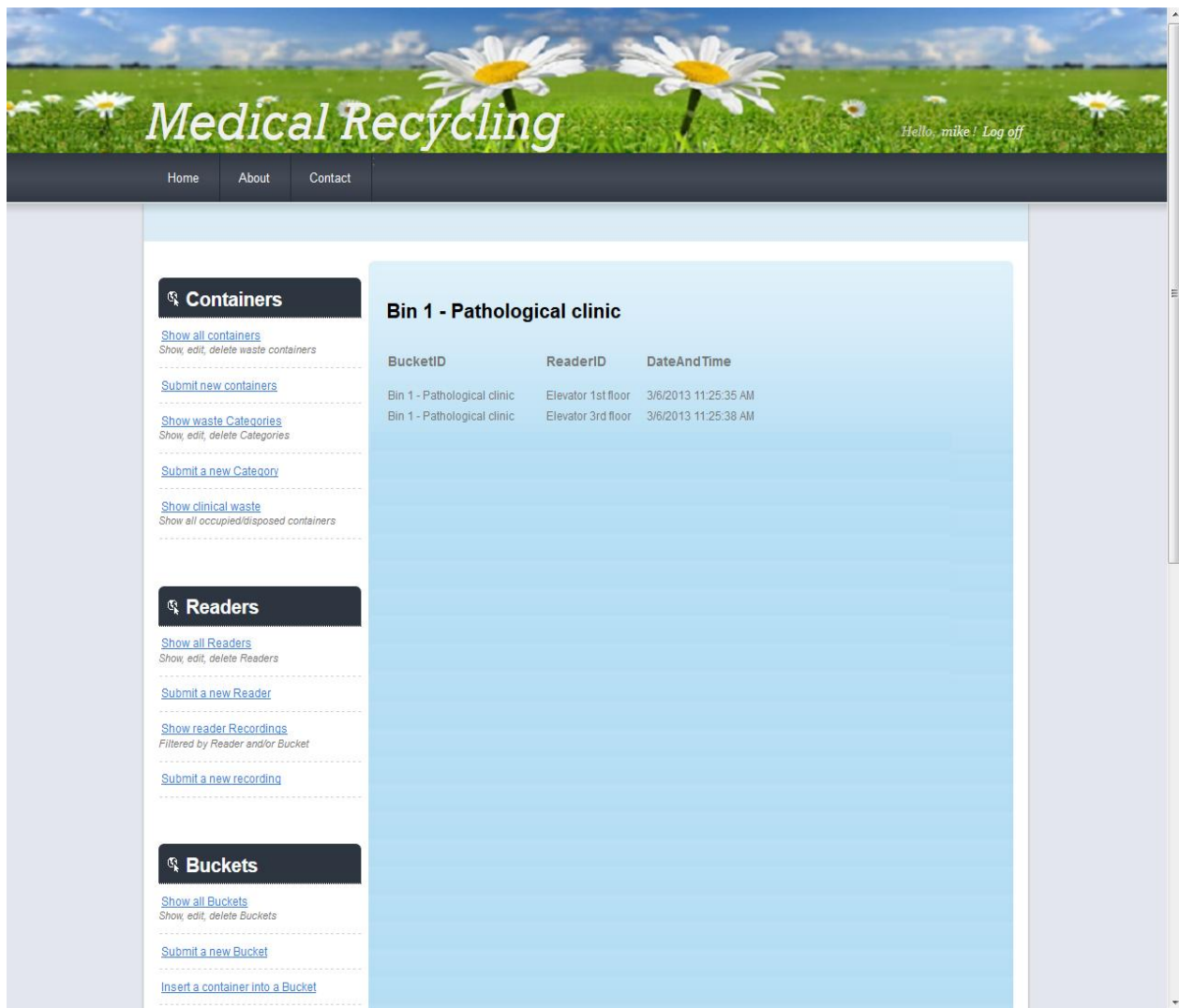
[Submit a new Bucket](#)

[Insert a container into a Bucket](#)

Elevator 1st floor

BucketID	ReaderID	DateAndTime
Bin 1 - Pathological clinic	First floor	3/6/2013 11:25:35 AM
Bin 1 - Pathological clinic	Third floor	3/6/2013 11:25:38 AM
Bin 1 - Neurological clinic	First floor	5/15/2013 3:06:25 AM
Trolley 1 - Surgery room A2	Basement	5/15/2013 3:06:45 AM
Cart 1 - Surgery	First floor	5/21/2013 1:29:08 PM
Infectious Clinic - Cart 1	First floor	5/21/2013 1:33:40 PM
Trolley 1 - Surgery room A2	First floor	5/21/2013 1:34:52 PM
Trolley 1 - Pathological clinic	First floor	5/21/2013 1:34:59 PM
Trolley 1 - Pathological clinic	First floor	5/21/2013 1:35:33 PM

Εικόνα 5.28. Εμφάνιση των καταγραφών του Reader Elevator – Frist floor.

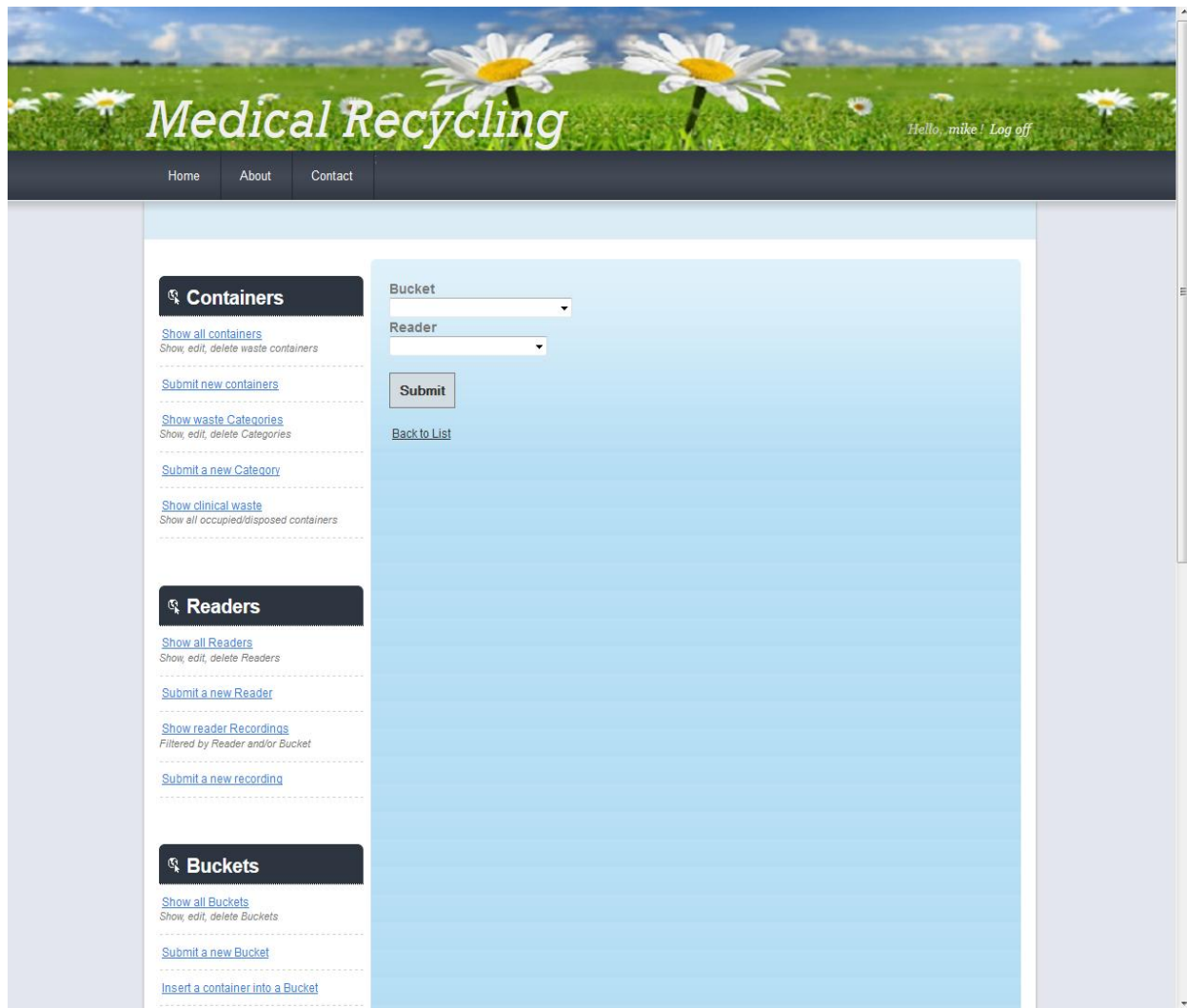


Εικόνα 5.29. Εμφάνιση των περιοχών (Αναγνώστες) που καταγράφηκε ο Συλλέκτης Bin 1 – Pathological Clinic.

5.5.4 Καταχώρηση νέας καταγραφής ενός Αναγνώστη

Στο σενάριο αυτό ο χρήστης καταχωρεί μία νέα καταγραφή ενός Συλλέκτη από κάποιον Αναγνώστη. Να σημειωθεί πως αυτή η λειτουργία τυπικά θα συνέβαινε αυτόματα, όταν ένας Reader αναγνώριζε το RFID tag ενός Συλλέκτη, αλλά καθώς δεν υπήρχε πρόσβαση σε τέτοιο εξοπλισμό σε κάποια νοσοκομειακή μονάδα, οι καταγραφές αυτές γίνονται εικονικά. Ο χρήστης καλείτε να επιλέξει μέσα από δύο Drop Down Lists έναν Συλλέκτη και τον Αναγνώστη ο οποίος τον κατέγραψε. Ως ημερομηνία καταγραφής θεωρείται (και αποθηκεύεται) η ώρα και ημερομηνία της καταχώρησης, όταν δηλαδή ο χρήστης πατάει το Submit. Για να χρησιμοποιήσει

ο χρήστης αυτήν την λειτουργία ακολουθεί το link *Submit a new recording* που βρίσκεται στο αντικείμενο Readers του sidebar. Μετά από επιτυχή καταχώρηση, η καταγραφή αυτή αποθηκεύεται στην βάση δεδομένων.

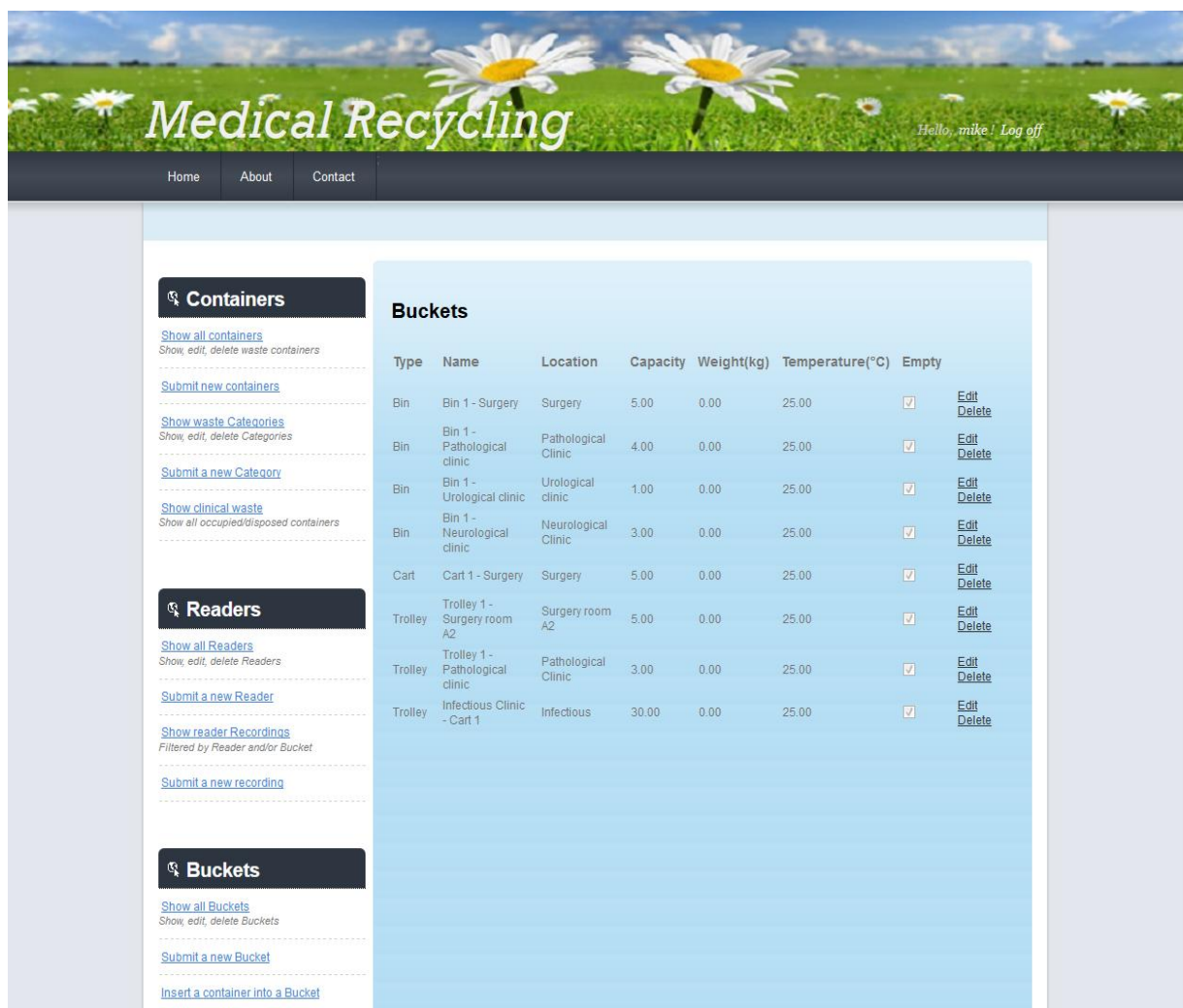


Εικόνα 5.30. Δημιουργία νέας καταγραφής Συλλέκτη από Αναγνώστη.

5.6 Διαχείριση των Συλλεκτών (Buckets)

5.6.1 Επισκόπηση, τροποποίηση και διαγραφή Συλλεκτών.

Στο σενάριο αυτό ο χρήστης επιλέγει να δει όλους τους καταχωρημένους συλλέκτες, και όλες τις πληροφορίες που σχετίζονται με αυτούς, στο σύστημα, και να τροποποιήσει ή να διαγράψει κάποιον από αυτούς, ακολουθώντας το link *Show all Buckets* στο αντικείμενο Buckets του sidebar.



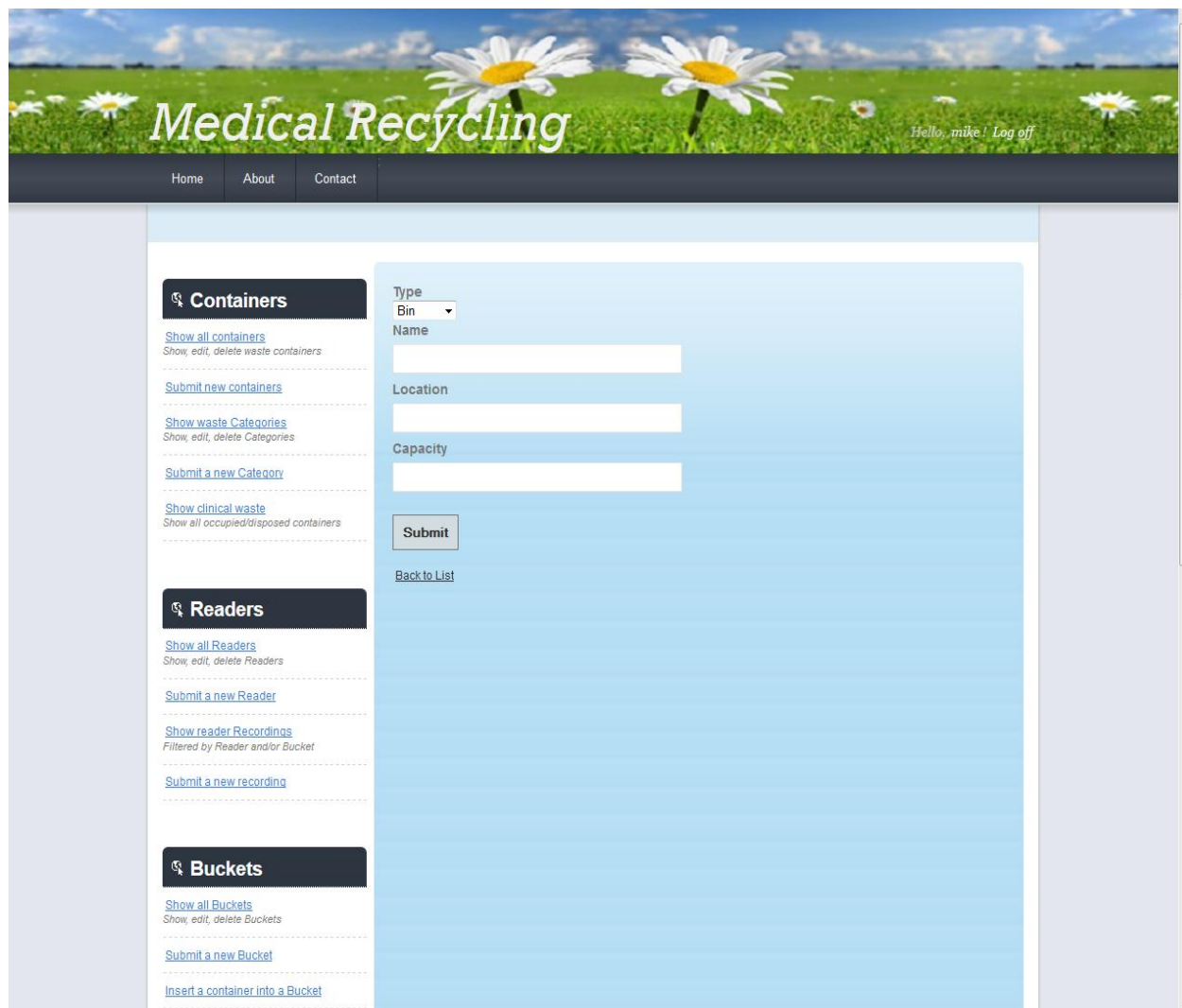
The screenshot shows the 'Medical Recycling' website interface. The header features a banner with daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into three sections: 'Containers', 'Readers', and 'Buckets'. The 'Buckets' section is active and displays a table of waste containers.

Type	Name	Location	Capacity	Weight(kg)	Temperature(°C)	Empty	
Bin	Bin 1 - Surgery	Surgery	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Pathological clinic	Pathological Clinic	4.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Urological clinic	Urological clinic	1.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Neurological clinic	Neurological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Cart	Cart 1 - Surgery	Surgery	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Surgery room A2	Surgery room A2	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Pathological clinic	Pathological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Infectious Clinic - Cart 1	Infectious	30.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete

Εικόνα 5.31. Συλλέκτες (Buckets).

5.6.2 Καταχώρηση νέου Συλλέκτη.

Σε αυτό το σενάριο ο χρήστης επιχειρεί να καταχωρήσει έναν νέο συλλέκτη, ακολουθώντας το link *Submit a new Bucket* που βρίσκεται στο αντικείμενο Buckets του sidebar. Αφού γίνει ο απαραίτητος έλεγχος στα απαραίτητα στοιχεία που καλείται να συμπληρώσει, και είναι επιτυχής, ο νέος Συλλέκτης καταχωρείται στο σύστημα.



The screenshot shows the 'Medical Recycling' website interface. At the top, there is a navigation bar with 'Home', 'About', and 'Contact' links. Below the navigation bar, there are three main sections: 'Containers', 'Readers', and 'Buckets'. Each section has a 'Show all' link and a 'Submit a new' link. The 'Buckets' section is highlighted, and the 'Submit a new Bucket' link is clicked, leading to a form. The form has a 'Type' dropdown menu set to 'Bin', and input fields for 'Name', 'Location', and 'Capacity'. A 'Submit' button is at the bottom of the form, and a 'Back to List' link is below it.

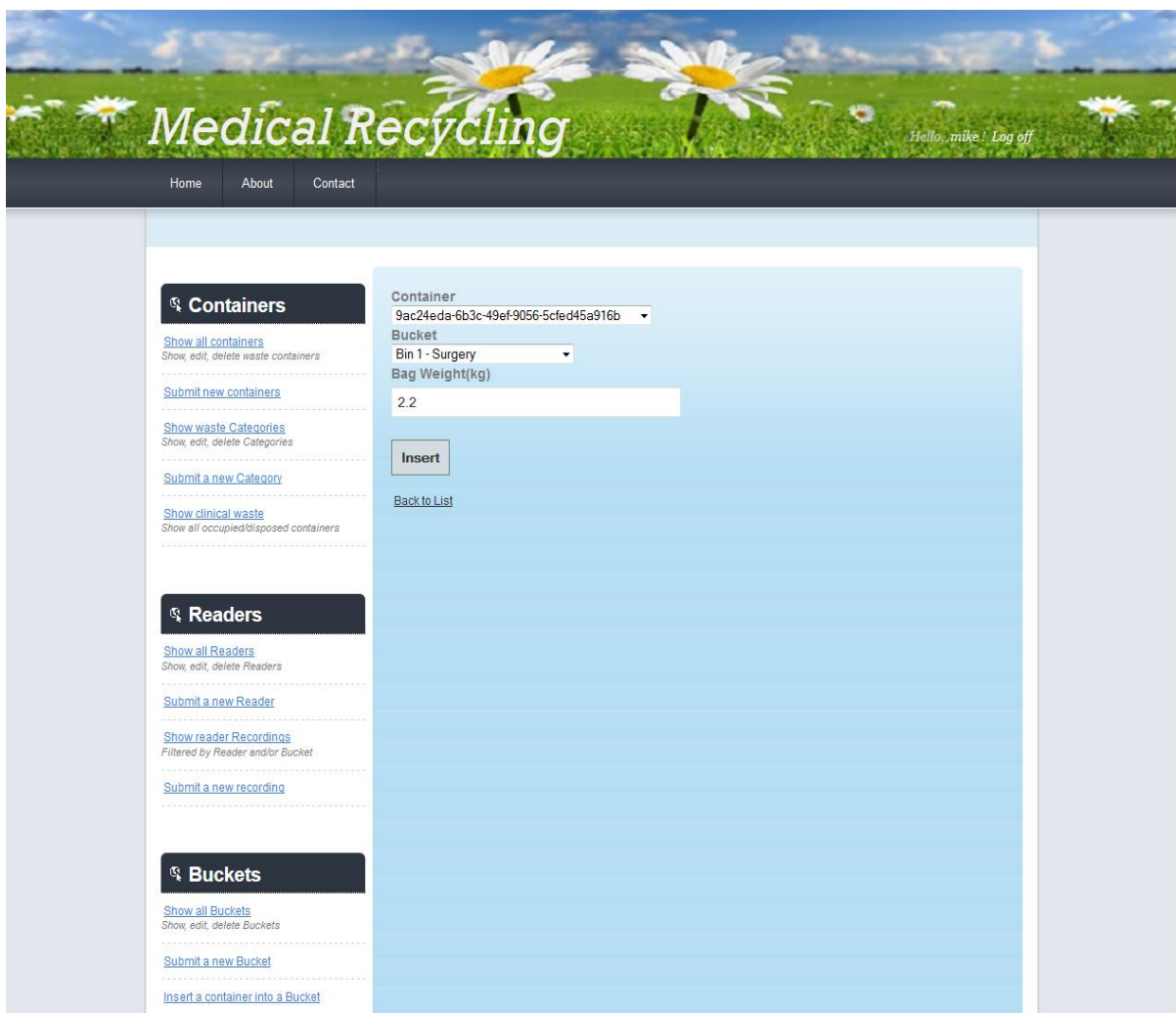
Εικόνα 5.32. Καταχώρηση νέου Συλλέκτη.

Πιο συγκεκριμένα, ο χρήστης καλείται μέσα από ένα Drop Down List να επιλέξει τον τύπο του Συλλέκτη (Bin – trolley – cart), να δώσει ένα όνομα για αυτόν, να συμπληρώσει την τοποθεσία στην οποία θα βρίσκεται, και την μέγιστη ονομαστική χωρητικότητα (σε κιλά) του. Το RFID tag του συλλέκτη καταχωρείται αυτόματα από το σύστημα, και η τρέχουσα θερμοκρασία τυπικά αποστέλλεται στην βάση δεδομένων δυναμικά από τους αισθητήρες, αλλά στην πράξη, επειδή δεν υπήρχε ο απαραίτητος εξοπλισμός, μπορούμε να μεταβάλλουμε την θερμοκρασία μόνο αλλάζοντας το σχετικό πεδίο μέσα από την βάση δεδομένων. Το τρέχον

βάρος κατά την καταχώρηση ενός νέου Συλλέκτη αρχικοποιείται στο 0. Όπως θα δούμε παρακάτω, κάθε φορά που εισέρχεται ένας καινούριος περιέκτης στον συλλέκτη, ανανεώνεται το βάρος του ανάλογα, και αντίστοιχα κατά την εκκένωσή του σε κάποιον χώρο προσωρινής αποθήκευσης το βάρος του μηδενίζεται. Αυτή η συμπεριφορά, που είναι και η αναμενόμενη, υλοποιήθηκε με χρήση Triggers [8] στην βάση δεδομένων του πληροφοριακού συστήματος.

5.6.3 Εισαγωγή ενός Περιέκτη σε έναν Συλλέκτη

Αυτό είναι ένα ακόμη σενάριο που πραγματοποιείται εικονικά λόγω της απουσίας πρόσβασης στον απαιτούμενο εξοπλισμό, κατά το οποίο ο χρήστης επιθυμεί να εισαγάγει έναν Περιέκτη σε έναν Συλλέκτη. Ο χρήστης καλείται να διαλέξει έναν περιέκτη μέσα από μια Drop Down List που εμφανίζεται, και εμφανίζει τα RFID tag των περιεκτών στο σύστημα. Να σημειωθεί πως εμφανίζονται μόνο οι περιέκτες οι οποίοι είναι άδειοι, καθώς δεν προβλέπεται τοποθέτηση απορριμμάτων σε περιέκτη που είναι ήδη κατειλημμένος (και συνεπώς βρίσκεται μέσα σε κάποιον Συλλέκτη). Αφού επιλεγεί ο περιέκτης, ο χρήστης συμπληρώνει το συνολικό βάρος του, και κατά την εισαγωγή του στον Συλλέκτη το βάρος αυτό προστίθεται στο τρέχων βάρος του Συλλέκτη (υλοποιήθηκε με Trigger στην βάση δεδομένων). Για να χρησιμοποιήσει αυτήν την λειτουργία ο χρήστης ακολουθεί το *Link Insert a container into a bucket* που βρίσκεται στο αντικείμενο Buckets του sidebar.



Εικόνα 5.33. Εισαγωγή ενός Περιέκτη σε έναν Συλλέκτη.

Στην παραπάνω εικόνα εισήχθη ο Περιέκτης με το RFID tag που φαίνεται στον Συλλέκτη Bin 1 – Surgery. Ο περιέκτης υποθέσαμε ότι είχε βάρος 2.2 kg. Όταν ο χρήστης πατήσει το κουμπί insert, μεταφέρεται στην λίστα με τους Συλλέκτες.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

[Show all containers](#)
Show, edit, delete waste containers

[Submit new containers](#)

[Show waste Categories](#)
Show, edit, delete Categories

[Submit a new Category](#)

[Show clinical waste](#)
Show all occupied/disposed containers

Buckets

Type	Name	Location	Capacity	Weight(kg)	Temperature(°C)	Empty	
Bin	Bin 1 - Surgery	Surgery	5.00	2.20	25.00	<input type="checkbox"/>	Edit Delete Details
Bin	Bin 1 - Pathological Clinic	Pathological Clinic	4.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Urological clinic	Urological clinic	1.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Neurological clinic	Neurological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Cart	Cart 1 - Surgery	Surgery	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Surgery room A2	Surgery room A2	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Pathological clinic	Pathological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Infectious Clinic - Cart 1	Infectious	30.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete

Readers

[Show all Readers](#)
Show, edit, delete Readers

[Submit a new Reader](#)

[Show reader Recordings](#)
Filtered by Reader and/or Bucket

[Submit a new recording](#)

Buckets

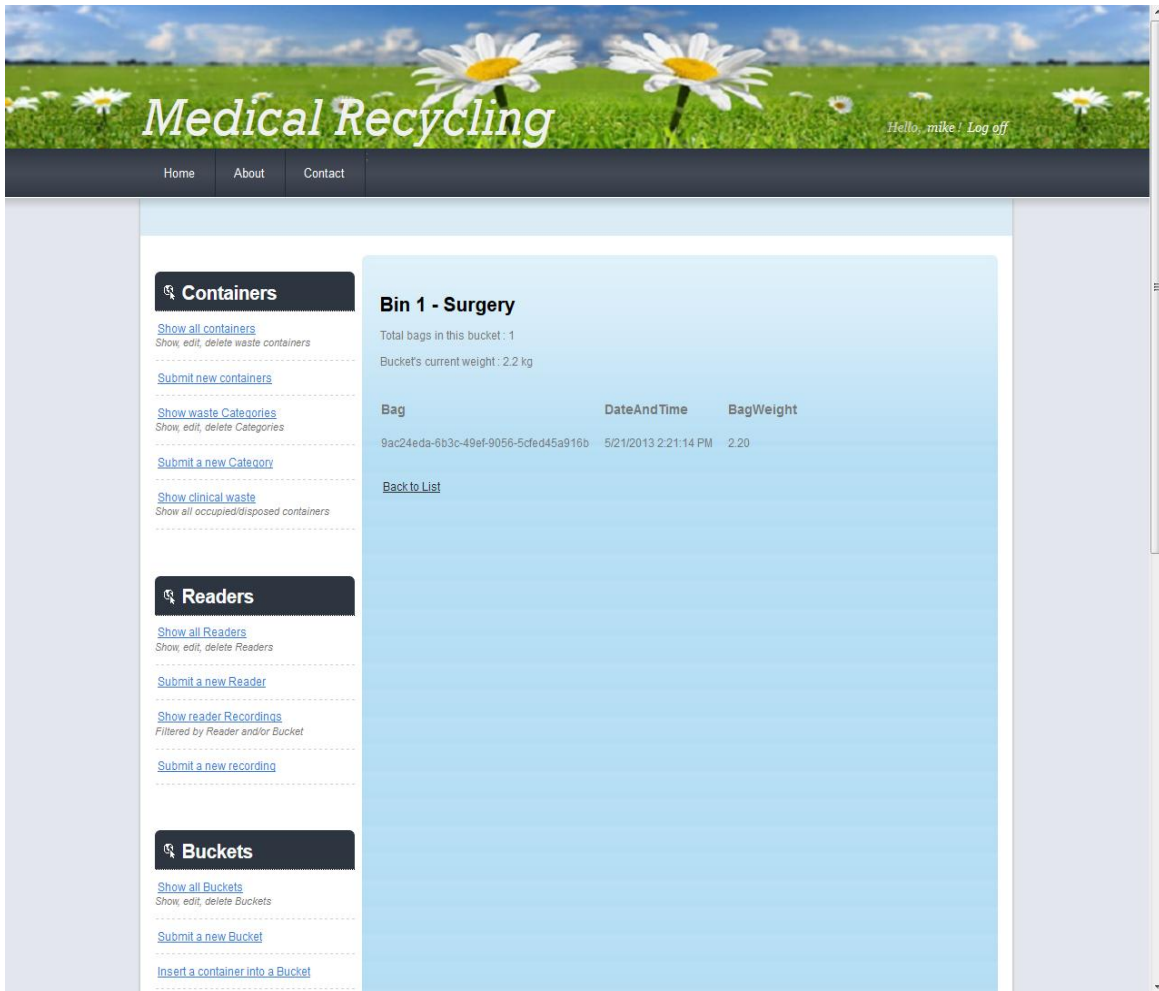
[Show all Buckets](#)
Show, edit, delete Buckets

[Submit a new Bucket](#)

[Insert a container into a Bucket](#)

Εικόνα 5.34. Συλλέκτης με περιέκτη

Βλέπουμε ότι στον Συλλέκτη Bin 1 – Surgery , όπου εισήχθηκε ο περιέκτης, έχει ανανεωθεί αυτόματα το βάρος του σε 2.2 kg , το checkbox που υποδηλώνει αν είναι άδειος δεν είναι πλέον επιλεγμένο, και έχει εμφανιστεί το Link *Details* , ενώ στους υπόλοιπους , άδειους Συλλέκτες δεν υπάρχει. Ακολουθώντας ο χρήστης το Link *Details* , μπορεί να δει ποιοι περιέκτες βρίσκονται στο εσωτερικό του συγκεκριμένου Συλλέκτη. Ακολουθώντας λοιπόν το συγκεκριμένο Link βλέπουμε:

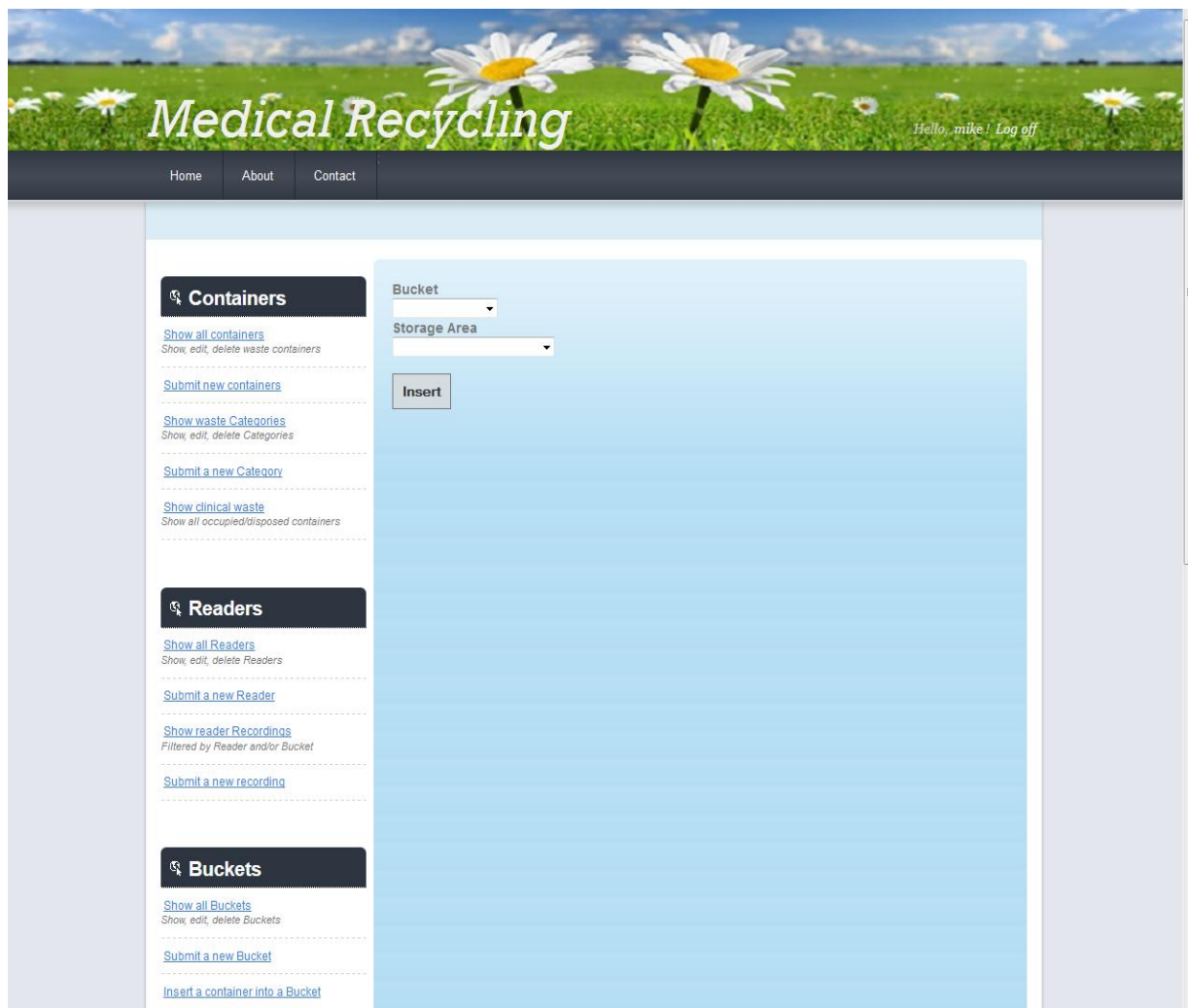


Εικόνα 5.35. Λεπτομέρειες Συλλέκτη.

Φαίνεται ο αριθμός των περιεκτών που βρίσκεται στο εσωτερικό του Συλλέκτη (στο παράδειγμα μας είναι 1), το συνολικό βάρος τους και ακολουθεί η λίστα με το RFID tag των περιεκτών, το βάρος του καθενός ξεχωριστά, και την ώρα και ημερομηνία που εισήλθε στον συγκεκριμένο Συλλέκτη.

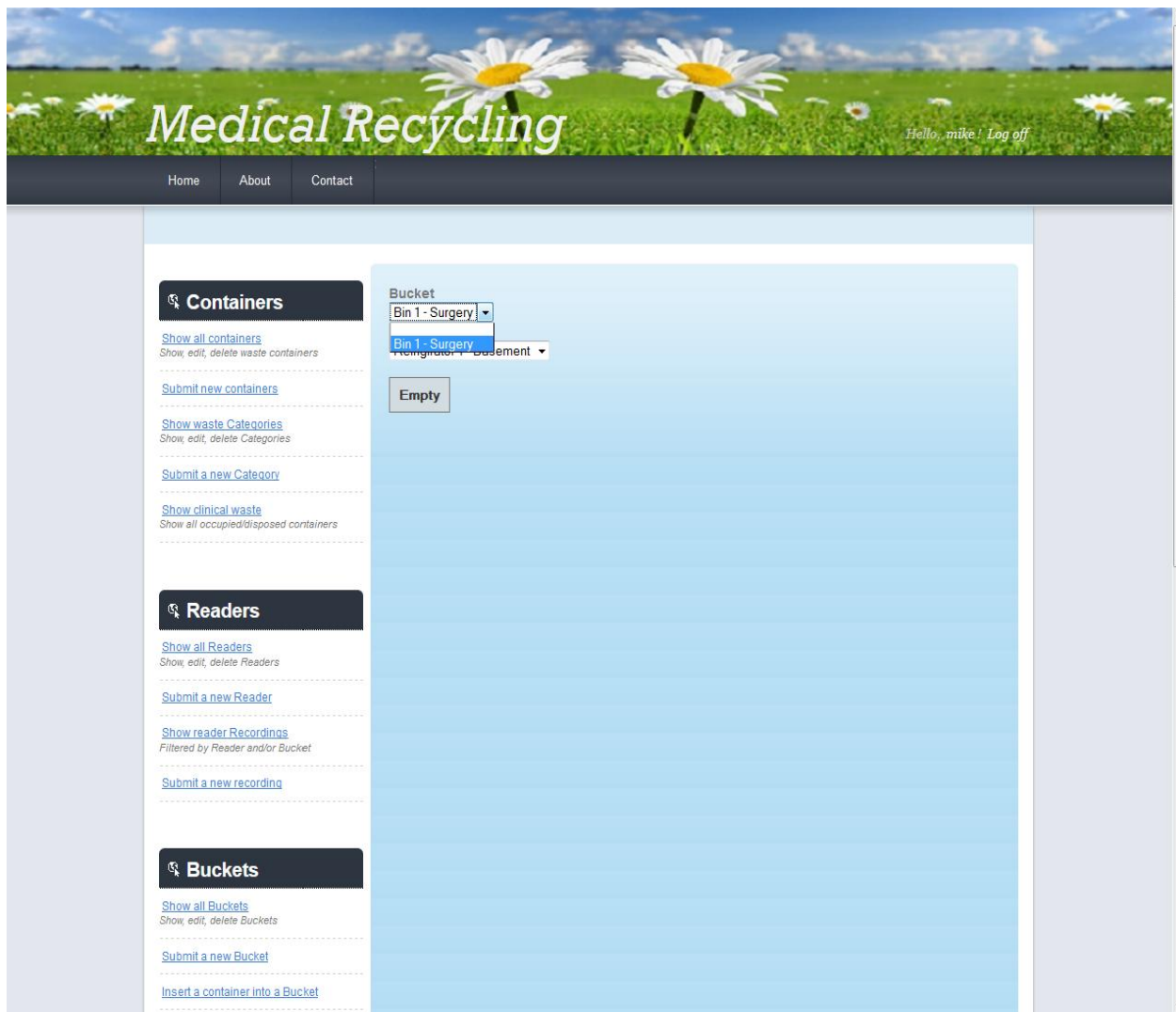
5.6.4 Εκκένωση ενός Συλλέκτη σε έναν Χώρο Προσωρινής Αποθήκευσης (Storage Area)

Σε αυτό το σενάριο, κατ' απόλυτη αναλογία με το προηγούμενο, ο χρήστης επιλέγει να εκκενώσει (εικονικά) έναν Συλλέκτη σε έναν χώρο προσωρινής αποθήκευσης. Η λειτουργία αυτή επιτυγχάνεται μέσω του συνδέσμου *Empty a Bucket into a Storeroom* που βρίσκεται στο αντικείμενο Buckets του Sidebar.



Εικόνα 5.36. Εκκένωση Συλλέκτη

Να σημειωθεί σε αυτή το σημείο, πως δεν έχει νόημα η εκκένωση ενός άδειου Συλλέκτη, συνεπώς στην Drop Down List όπου καλείται ο χρήστης να επιλέξει Συλλέκτη εμφανίζονται μόνο οι Συλλέκτες οι οποίοι έχουν κάποιον Περιέκτη στο εσωτερικό τους. Στο παράδειγμά μας, ο μοναδικός τέτοιος Συλλέκτης είναι ο Bin 1 – Surgery, που χρησιμοποιήσαμε στο προηγούμενο παράδειγμα.



Εικόνα 5.37. Εκκένωση Συλλέκτη.

Όταν ο χρήστης πιάσει το κουμπί *Empty* ο συγκεκριμένος συλλέκτης εκκενώνεται σε κάποιον χώρο προσωρινής αποθήκευσης, στο παράδειγμά μας στο ψυγείο Refrigerator 1 – Basement, και μεταφέρεται ξανά στην λίστα με τους συλλέκτες, όπου ο συγκεκριμένος εμφανίζεται ξανά ως άδειος και με μηδενικό βάρος. Η συγκεκριμένη συμπεριφορά, που είναι και η αναμενόμενη, υλοποιήθηκε στην βάση δεδομένων με χρήση Triggers.

The screenshot shows a web application interface for 'Medical Recycling'. The header features a banner with daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into three sections: 'Containers', 'Readers', and 'Buckets'. The 'Buckets' section contains a table with the following data:

Type	Name	Location	Capacity	Weight(kg)	Temperature(°C)	Empty	
Bin	Bin 1 - Surgery	Surgery	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Pathological Clinic	Pathological Clinic	4.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Urological clinic	Urological clinic	1.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Bin	Bin 1 - Neurological clinic	Neurological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Cart	Cart 1 - Surgery	Surgery	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Surgery room A2	Surgery room A2	5.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Trolley 1 - Pathological Clinic	Pathological Clinic	3.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete
Trolley	Infectious Clinic - Cart 1	Infectious	30.00	0.00	25.00	<input checked="" type="checkbox"/>	Edit Delete

Εικόνα 5.38. Εκκένωση Συλλέκτη – Μηδενισμός βάρους.

5.7 Διαχείριση των Χώρων Προσωρινής Αποθήκευσης (Storage Areas)

5.7.1 Επισκόπηση, τροποποίηση και διαγραφή Χώρων Προσωρινής Αποθήκευσης

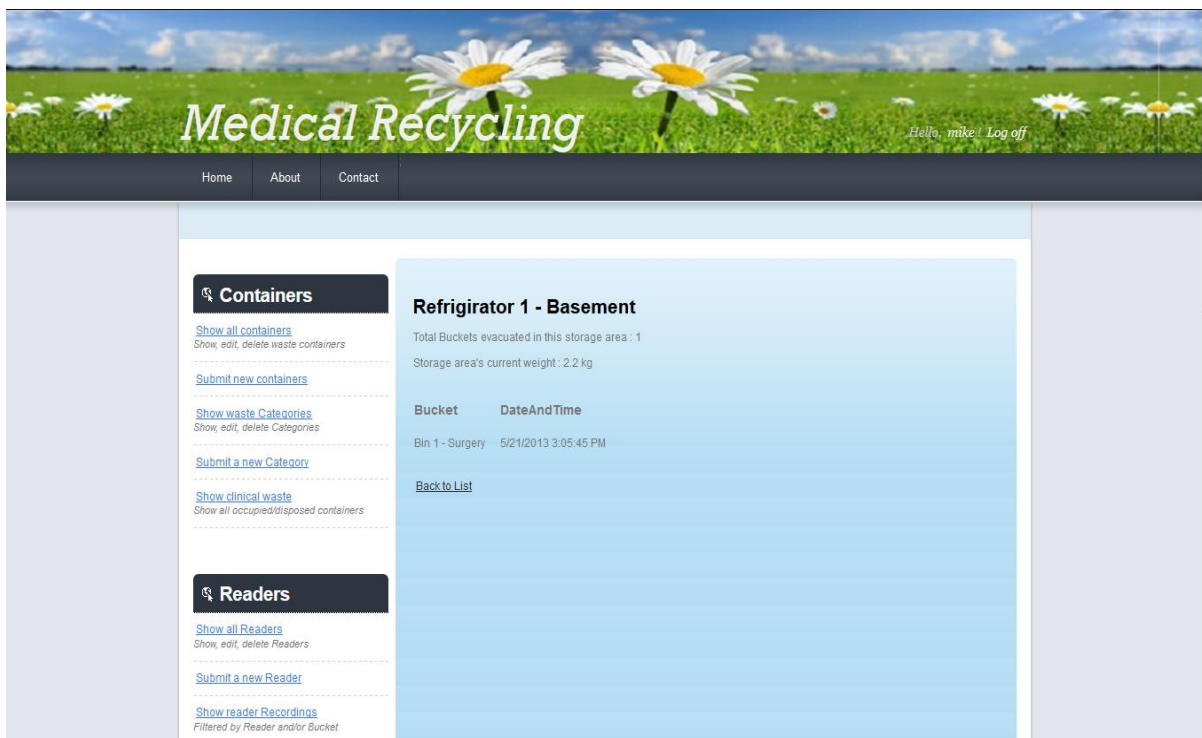
Στο σενάριο αυτό ο χρήστης επιχειρεί να δει τους χώρους προσωρινής αποθήκευσης που είναι καταχωρημένοι στην βάση δεδομένων, μαζί με όλες τις πληροφορίες που τους συνοδεύουν, και να τροποποιήσει ή να διαγράψει κάποιον από αυτούς, ακολουθώντας το Link *Show all Storage Areas* που βρίσκεται στο αντικείμενο Storage Areas του sidebar.

The screenshot shows the 'Medical Recycling' website. At the top, there is a banner with the title 'Medical Recycling' and a user greeting 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into two sections: 'Containers' and 'Readers'. The 'Containers' section has several links: 'Show all containers', 'Submit new containers', 'Show waste Categories', 'Submit a new Category', and 'Show clinical waste'. The 'Readers' section has links: 'Show all Readers', 'Submit a new Reader', 'Show reader Recordings', and 'Submit a new recording'. The 'Storage Areas' section contains a table with the following data:

Type	Name	Location	Capacity(kg)	Weight(kg)	Temperature(°C)	
Refrigrator	Refrigrator 1 - Basement	Basement	50.00	2.20	3.00	Edit Delete Details
Refrigrator	Refrigrator 2 - Basement	Basement	60.00	0.00	3.00	Edit Delete
Storeroom	Storeroom 1 - Basement	Basement	50.00	0.00	24.00	Edit Delete
Storeroom	Storeroom 2 - Basement	Basement	100.00	0.00	24.00	Edit Delete
Storeroom	Storeroom - First floor	1st Floor	100.00	0.00	24.00	Edit Delete

Εικόνα 5.39. Χώροι Προσωρινής Αποθήκευσης

Βλέπουμε ότι, κατ' απόλυτη αναλογία με τους Συλλέκτες στην περίπτωση που κάποιος χώρος προσωπικής αποθήκευσης δεν είναι άδειος, αλλά περιέχει απόβλητα, εμφανίζεται το Link *Details*. Ακολουθώντας το εμφανίζονται λεπτομέρειες όπως ποιοι Συλλέκτες εκκενώθηκαν σε αυτών και την ώρα και ημερομηνία που αυτό συνέβη. Στο παράδειγμα μας στην προηγούμενη ενότητα, εκκενώσαμε τον κάδο Bin 1 – Surgery στο ψυγείο Refrigerator 1 – Basement, και βλέπουμε ότι το καθαρό βάρος του ψυγείου αντιστοιχεί στο βάρος του Συλλέκτη όταν εκκενώθηκε σε αυτό. Αν ο χρήστης ακολουθήσει το σύνδεσμο *Details* βλέπει αυτό που περιγράψαμε.



Εικόνα 5.40. Χώροι Προσωρινής Αποθήκευσης - Λεπτομέρειες

5.7.2 Καταχώρηση νέου χώρου προσωρινής αποθήκευσης

Σε αυτό το σενάριο ο χρήστης επιχειρεί να καταχωρήσει έναν νέο χώρο προσωρινής αποθήκευσης, ακολουθώντας το link *Submit a new Storage Area* που βρίσκεται στο αντικείμενο *Storage Areas* του sidebar. Αφού γίνει ο απαραίτητος έλεγχος στα απαραίτητα στοιχεία που καλείται να συμπληρώσει, και είναι επιτυχής, ο νέος χώρος προσωρινής αποθήκευσης καταχωρείται στο σύστημα.

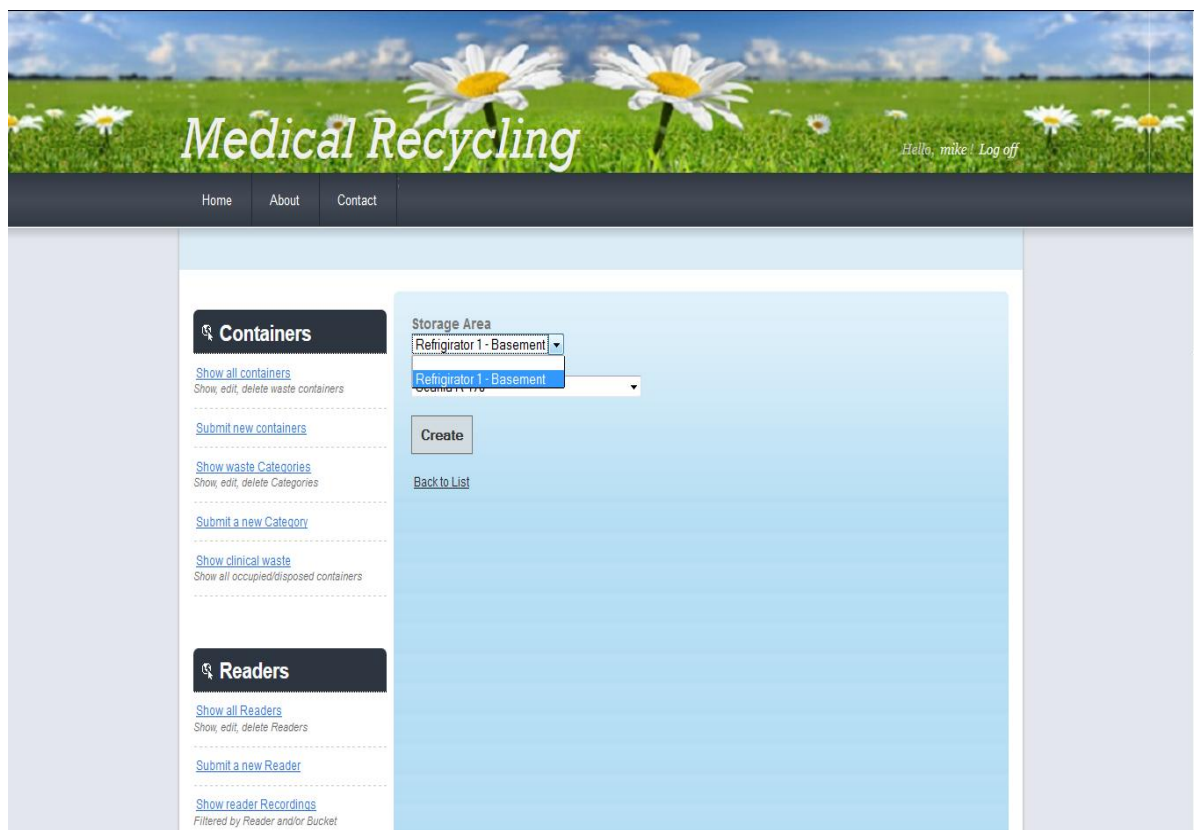


Εικόνα 5.41. Καταχώρηση νέου χώρου προσωρινής αποθήκευσης.

Πιο συγκεκριμένα, ο χρήστης καλείται μέσα από ένα Drop Down List να επιλέξει τον τύπο του χώρου προσωρινής αποθήκευσης (Storeroom – Refrigerator – Lockable Cupboard), να δώσει ένα όνομα για αυτόν, να συμπληρώσει την τοποθεσία στην οποία θα βρίσκεται, και την μέγιστη ονομαστική χωρητικότητα (σε κιλά) του. Η τρέχουσα θερμοκρασία τυπικά αποστέλλεται στην βάση δεδομένων δυναμικά από τους αισθητήρες, αλλά στην πράξη, επειδή δεν υπήρχε ο απαραίτητος εξοπλισμός, μπορούμε να μεταβάλλουμε την θερμοκρασία μόνο αλλάζοντας το σχετικό πεδίο μέσα από την βάση δεδομένων. Το τρέχον βάρος κατά την καταχώρηση ενός νέου χώρου προσωρινής αποθήκευσης αρχικοποιείται στο μηδέν και κατά τα γνωστά ανανεώνεται αυτόματα κατά την εκκένωση του σε κάποιο όχημα μεταφοράς (μηδενίζεται) ή κατά την εκκένωση κάποιου Συλλέκτη σε αυτόν (προστίθεται το τρέχον βάρος του Συλλέκτη).

5.7.3 Εκκένωση ενός χώρου προσωρινής αποθήκευσης σε κάποιο όχημα μεταφοράς

Σε αυτό το σενάριο ο χρήστης επιλέγει να εκκενώσει (εικονικά) έναν χώρο προσωρινής αποθήκευσης σε κάποιο όχημα μεταφοράς. Η λειτουργία αυτή επιτυγχάνεται μέσω του συνδέσμου *Empty a Storage Area into a Vehicle* που βρίσκεται στο αντικείμενο Storage Areas του Sidebar.



Εικόνα 5.42. Εκκένωση χώρου προσωρινής αποθήκευσης.

Να σημειωθεί σε αυτή το σημείο, πως δεν έχει νόημα η εκκένωση ενός άδειου χώρου προσωρινής αποθήκευσης, συνεπώς στην Drop Down List όπου καλείται ο χρήστης να επιλέξει χώρο προσωρινής αποθήκευσης εμφανίζονται μόνο εκείνοι οι οποίοι περιέχουν απόβλητα στο εσωτερικό τους. Στο παράδειγμά που τρέξαμε στις προηγούμενες ενότητες, ο μοναδικός τέτοιος χώρος είναι ο Refrigerator 1 – Basement , συνεπώς μόνο αυτός εμφανίζεται. Αν επιλέξει ο χρήστης κάποιο όχημα μεταφοράς, στο παράδειγμά μας επιλέξαμε το όχημα Mercedes-Benz 770 και αδειάζει τον χώρο προσωρινής αποθήκευσης, γίνονται όλες οι απαραίτητες ενέργειες (μηδενίζεται το βάρος του και διαγράφονται οι καταχωρήσεις που αφορούν Συλλέκτες που έχουν εκκενωθεί σε αυτόν).

The screenshot shows a web application interface for 'Medical Recycling'. At the top, there is a navigation bar with 'Home', 'About', and 'Contact' links. Below the navigation bar, there are two main sections: 'Containers' and 'Readers'. The 'Containers' section includes links for 'Show all containers', 'Submit new containers', 'Show waste Categories', 'Submit a new Category', and 'Show clinical waste'. The 'Readers' section includes links for 'Show all Readers', 'Submit a new Reader', and 'Show reader Recordings'. The central part of the page features a 'Storage Areas' table with the following data:

Type	Name	Location	Capacity(kg)	Weight(kg)	Temperature(°C)	
Refrigerator	Refrigerator 1 - Basement	Basement	50.00	0.00	3.00	Edit Delete
Refrigerator	Refrigerator 2 - Basement	Basement	60.00	0.00	3.00	Edit Delete
Storeroom	Storeroom 1 - Basement	Basement	50.00	0.00	24.00	Edit Delete
Storeroom	Storeroom 2 - Basement	Basement	100.00	0.00	24.00	Edit Delete
Storeroom	Storeroom - First floor	1st Floor	100.00	0.00	24.00	Edit Delete

Εικόνα 5.43. Χώροι προσωρινής αποθήκευσης μετά την εκκένωση

5.8 Διαχείριση των Οχημάτων Μεταφοράς (Vehicles) και των Οδηγών (Drivers)

5.8.1 Επισκόπηση, τροποποίηση και διαγραφή οχημάτων μεταφοράς και οδηγών

Σε αυτό το σενάριο ο χρήστης επιχειρεί να δει όλα τα καταχωρημένα οχήματα μεταφοράς και τους διαθέσιμους οδηγούς, και όλων των απαραίτητων στοιχείων που σχετίζονται με αυτούς και να τροποποιήσει κάποιο στοιχείο τους ή να διαγράψει κάποιον από αυτούς. Για να χρησιμοποιήσει αυτήν την λειτουργία ακολουθεί τα link *Show all Drivers* και *Show all Vehicles* που βρίσκεται στο αντικείμενο Schedules του sidebar.

The screenshot shows the 'Medical Recycling' website with a header featuring a field of daisies. The navigation menu includes 'Home', 'About', and 'Contact'. The main content area is divided into two sections: 'Containers' and 'Readers'. The 'Containers' section has links for 'Show all containers', 'Submit new containers', 'Show waste Categories', 'Submit a new Category', and 'Show clinical waste'. The 'Readers' section has links for 'Show all Readers', 'Submit a new Reader', and 'Show reader Recordings'. The 'Drivers' table is displayed in the center, listing driver information with 'Edit' and 'Delete' links for each row.

FirstName	LastName	PhoneNumber	Address	EmailAddress	
Giorgos	Printezis	699999999	Xrissipoy 4, Zografou	gprintezis@gmail.com	Edit Delete
Kostas	Papanikolaou	698888888	Dabaki Plindou 44, Goudi	kostaspapanikolaou@gmail.com	Edit Delete
Vasilis	Spanoulis	697777777	Diogenous 35, Korydallos	killbill@gmail.com	Edit Delete
Kostas	Sloukas	696666666	Kousidi 33, Zografou	ksloukas@gmail.com	Edit Delete
Kyle	Hines	695555555	Ηλιοτρόπιη, Αγρυμνίου 6	kyehines@gmail.com	Edit Delete
Stratos	Perperoglou	693333333	Ektoros 23, Korydallos	startosperp@gmail.com	Edit Delete

Εικόνα 5.44. Οδηγοί.

The screenshot shows the 'Medical Recycling' website with the same header and navigation menu as the previous image. The 'Containers' and 'Readers' sections are visible on the left. The 'Trucks' table is displayed in the center, listing truck information with 'Edit', 'Delete', and 'Details' links for each row.

Name	Capacity	CurrentWeight	CurrentTemperature	
Scania R 470	1000.00	0.00	25.00	Edit Delete
Scania R 560 Fire	1200.00	0.00	25.00	Edit Delete
Scania R 620 Garbage	1400.00	0.00	25.00	Edit Delete
Mercedes-Benz 770	1300.00	2.20	25.00	Edit Delete Details
Mercedes-Benz AXOR 2633 6X4 MIXER	17000.00	0.00	25.00	Edit Delete

Εικόνα 5.45. Οχήματα.

Παρατηρούμε ότι στο όχημα Mercedes-Benz 770, όπου χρησιμοποιήσαμε στην προηγούμενη ενότητα για να εκκενώσουμε το ψυγείο Refrigerator 1 – Basement, εμφανίζεται ο σύνδεσμος *Details* και αν ο χρήστης τον ακολουθήσει βλέπει πληροφορίες όπως ποιοι χώροι προσωρινής

αποθήκευσης εκκενώθηκαν στο συγκεκριμένο όχημα (στο παράδειγμά μας εμφανίζεται το ψυγείο Refrigerator 1 – Basement , και την ώρα και ημερομηνία που αυτό συνέβη).



Εικόνα 5.46. Οχήματα – Λεπτομέρειες.

Παρατηρούμε ότι υπάρχει και το πεδίο RouteID που αντιστοιχεί στον κωδικό της συγκεκριμένης διαδρομής. Το όχημα στο παράδειγμά μας δεν έχει ξεκινήσει την διαδρομή. Απλώς έχει φορτωθεί τα απορρίμματα από έναν χώρο προσωρινής αποθήκευσης. Στο επόμενο σενάριο θα στείλουμε το συγκεκριμένο όχημα στον προορισμό διαχείρισης των αποβλήτων, οπότε στο πεδίο Route θα καταχωρηθεί ο κωδικός της διαδρομής.

5.8.2 Καταχώρηση νέου οχήματος μεταφοράς και οδηγού

Σε αυτό το σενάριο ο χρήστης επιχειρεί να καταχωρήσει έναν νέο οδηγό ή όχημα μεταφοράς, ακολουθώντας τα link *Submit a new Driver* και *Submit a new Vehicle* που βρίσκεται στο αντικείμενο Schedules του sidebar. Αφού γίνει ο απαραίτητος έλεγχος στα απαραίτητα στοιχεία που καλείται να συμπληρώσει, και είναι επιτυχής, ο νέος οδηγός ή το νέο όχημα μεταφοράς καταχωρείται στην βάση δεδομένων του συστήματος.



Εικόνα 5.47. Καταχώρηση νέου οδηγού.

5.8.3 Διαχείριση δρομολογίων

Σε αυτό το σενάριο ο χρήστης επιχειρεί να δει όλα τα δρομολόγια των οχημάτων μεταφοράς που είτε έχουν ολοκληρωθεί είτε αναμένεται να ολοκληρωθούν, ακολουθώντας το Link *Show all Itineraries* που βρίσκεται στο αντικείμενο *Storage Areas Schedules* του sidebar.

The screenshot shows the 'Medical Recycling' website. At the top, there is a banner with daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. Below the banner is a navigation menu with 'Home', 'About', and 'Contact'. The main content area is divided into a sidebar on the left and a main content area on the right. The sidebar contains two sections: 'Containers' and 'Readers'. The 'Containers' section has links for 'Show all containers', 'Submit new containers', 'Show waste Categories', 'Submit a new Category', and 'Show clinical waste'. The 'Readers' section has links for 'Show all Readers', 'Submit a new Reader', and 'Show reader Recordings'. The main content area features a table titled 'Itineraries' with columns for 'Driver', 'Truck', 'Date and Time', and 'Finished'.

Driver	Truck	Date and Time	Finished
Printzis	Scania R 470	4/12/2013 12:00:00 AM	<input checked="" type="checkbox"/>
Stoukas	Scania R 470	4/17/2013 12:00:00 AM	<input checked="" type="checkbox"/>
Spanoulis	Scania R 560 Fire	5/16/2013 12:29:12 AM	<input checked="" type="checkbox"/>
Hines	Scania R 470	5/16/2013 12:49:59 PM	<input checked="" type="checkbox"/>

Εικόνα 5.48. Δρομολόγια.

Ο χρήστης μπορεί να καταχωρήσει ένα νέο δρομολόγιο ακολουθώντας τον σύνδεσμο *Submit a new Itinerary*, που βρίσκεται στο αντικείμενο Schedules του sidebar. Ο χρήστης καλείται να επιλέξει μέσα από Drop Down List, ένα διαθέσιμο όχημα και έναν διαθέσιμο οδηγό.



Εικόνα 5.49. Καταχώρηση νέου δρομολογίου.

Να σημειωθεί πως έχει νόημα να αρχίσει ένα καινούριο δρομολόγιο, μόνο αν στο όχημα που θα επιλεγεί έχουν φορτωθεί απόβλητα, και προφανώς ο οδηγός αυτού θα πρέπει να είναι ελεύθερος, και να μην εκτελεί κάποιο άλλο δρομολόγιο εκείνη την στιγμή. Έτσι στην συγκεκριμένη λίστα εμφανίζεται μόνο το όχημα Mercedes-Benz 770 που χρησιμοποιήθηκε για την φόρτωση στην προηγούμενη ενότητα, ενώ όλοι οι οδηγοί είναι διαθέσιμοι.

Όταν ο χρήστης πατήσει το κουμπί submit ώστε να ξεκινήσει το δρομολόγιο, αυτό εμφανίζεται στην λίστα με τα δρομολόγια ως ενεργό.

The screenshot shows the 'Medical Recycling' website. The header features a banner with daisies and the text 'Medical Recycling' and 'Hello, mike! Log off'. A navigation bar contains 'Home', 'About', and 'Contact'. The main content area is divided into two columns. The left column has two sections: 'Containers' and 'Readers', each with links for showing, editing, deleting, and submitting new items. The right column is titled 'Itineraries' and contains a table with the following data:

RouteID	Driver	Truck	Date and Time	Finished
1	Printezis	Scania R 470	4/12/2013 12:00:00 AM	<input checked="" type="checkbox"/>
2	Sloukas	Scania R 470	4/17/2013 12:00:00 AM	<input checked="" type="checkbox"/>
3	Spanoulis	Scania R 560 Fire	5/16/2013 12:29:12 AM	<input checked="" type="checkbox"/>
4	Hines	Scania R 470	5/16/2013 12:49:59 PM	<input checked="" type="checkbox"/>
5	Spanoulis	Mercedes-Benz 770	5/21/2013 7:32:21 PM	<input type="checkbox"/> Mark route as finished

Εικόνα 5.50. Ενεργό δρομολόγιο

Παρατηρούμε ότι έχει δημιουργηθεί ο σύνδεσμος *Mark Route as Finished*, που μπορεί να χρησιμοποιήσει ο χρήστης για να ολοκληρώσει εικονικά το συγκεκριμένο δρομολόγιο.

Επιστρέφοντας στην λίστα με τα οχήματα, αν ο χρήστης επιλέξει τον σύνδεσμο *Details* στο όχημα Mercedes-Benz 770, βλέπουμε ότι έχει ανανεωθεί το πεδίο RouteID, και έχει την τιμή 5, που αντιστοιχεί στο δρομολόγιο που μόλις καταχωρήθηκε.



Εικόνα 5.51. Όχημα – λεπτομέρειες.

Αν ο χρήστης επιλέξει να τερματίσει το δρομολόγιο (*Mark Route as finished*) κατά τα γνωστά μηδενίζεται το βάρος του οχήματος μεταφοράς, και σβήνονται οι καταχωρήσεις των χώρων προσωρινής αποθήκευσης (στο παράδειγμά μας του Refrigerator 1 – Basement) σε αυτό.

5.9 Ειδοποιήσεις (Notifications)

Το πληροφοριακό σύστημα ελέγχει τους περιορισμούς που επιβάλλονται από τις διαδικασίες διαχείρισης των Ι.Α , και ενημερώνει τον χρήστη όταν κάποιος κανόνας παραβιάζεται. Εκτελούνται συνεχώς έλεγχοι και όταν κριθεί σκόπιμο ενημερώνεται ο χρήστης, ότι ενέργεια και σε όποιο View και να βρίσκεται στον ιστότοπό μας.

Πιο συγκεκριμένα ελέγχονται συνεχώς μέσω SQL Jobs [12] οι ημερομηνίες λήξης από τους περιέκτες, ο μέγιστος χρόνος που επιτρέπεται να μείνουν τα απόβλητα στους Συλλέκτες πριν μεταφερθούν στους χώρους προσωρινής αποθήκευσης, και οι θερμοκρασίες σε αυτούς τους χώρους. Αν κάποιος από τους παραπάνω κανόνες παραβιάζεται, δημιουργείται μια νέα εγγραφή στον πίνακα Notifications της βάσης δεδομένων, και εμφανίζεται popup window στον χρήστη στο πάνω μέρος της οθόνης, καθώς επίσης και ένα νέο menu item *Notifications* με κίτρινο χρώμα, που υποδηλώνει ότι υπάρχουν ειδοποιήσεις προς τον χρήστη.

Ας δούμε τι συμβαίνει για παράδειγμα όταν φτάνει η ημερομηνία λήξης μερικών περιεκτών.

Medical Recycling

Notifications(1) Home About Contact

Containers

Show all containers
Show, edit, delete waste containers

Submit new containers

Show waste Categories
Show, edit, delete Categories

Submit a new Category

Show clinical waste
Show all occupied/disposed containers

Readers

Show all Readers
Show, edit, delete Readers

Submit a new Reader

Show reader Recordings
Filtered by Reader and/or Bucket

Containers

Group by: RFIDtag | Category | Date Arrived

Type	RFIDtag	Date Arrived	Date Expires	Size	Empty	Category	Color	
Plastic Bag	d860f83c-d2cf-460-327f597e...	3/6/2013	11/11/2015	5	<input type="checkbox"/>	IA-AX	Red	Edit Delete
Plastic Bag	aa4d796-a070-522...	3/6/2013	11/11/2015	5	<input type="checkbox"/>	IA-AX	Red	Edit Delete
Plastic Bag	0e9f2aa5-b0033a6e...	3/6/2013	11/1/2014	4	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	1ab1f7bb-8860-405...	3/18/2013	12/22/2014	7	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	0a3480f8-41eb-bd5f-6388a430...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	b18ee617-a097-4376...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	48ba65e-ba73-4ed-a6ec-2f29...	3/18/2013	12/12/2014	11	<input checked="" type="checkbox"/>	AIA	Red	Edit Delete
Plastic Bag	5a61aae...							

Εικόνα 5.52. Notification

Αν ο χρήστης πατήσει το menu item notification, οδηγείται στο ακόλουθο View.

Medical Recycling

Notifications(1) Home About Contact

Containers

Show all containers
Show, edit, delete waste containers

Submit new containers

Show waste Categories
Show, edit, delete Categories

Submit a new Category

Show clinical waste
Show all occupied/disposed containers

Readers

Show all Readers
Show, edit, delete Readers

Submit a new Reader

Show reader Recordings
Filtered by Reader and/or Bucket

Notifications

type info

Bags There are currently 110 Bags that are expired or going to expire soon [Show Bags](#)

Εικόνα 5.53. Notification View.

Ο χρήστης ενημερώνεται για το γεγονός ότι σε μερικούς περιέκτες έχει παρέλθει η ημερομηνία

λήξης. Αν ακολουθήσει τον σύνδεσμο *Show Containers* δίπλα από το ενημερωτικό μήνυμα, μεταφέρεται στην σελίδα με αυτούς, και διαπιστώνει ότι μερικοί έχουν λήξει. Για γρήγορη αναγνώριση εκείνοι που έχουν λήξει είναι σημειωμένοι με κόκκινο χρώμα.

The screenshot shows the 'Medical Recycling' website interface. At the top, there is a navigation bar with links for 'Notifications(1)', 'Home', 'About', and 'Contact'. Below this, there are sections for 'Containers' and 'Readers'. The 'Containers' section contains several links: 'Show all containers', 'Submit new containers', 'Show waste Categories', 'Submit a new Category', and 'Show clinical waste'. The 'Readers' section contains links for 'Show all Readers', 'Submit a new Reader', and 'Show reader Recordings'. The main content area displays a table of containers with the following data:

Count	Type	CATEGORY	DateArrived	DateExpires	Size	Color
20	Hospital box	EIA-MTX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM	5	-
100	Plastic Bag	IA-AX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM	3	Black
100	Plastic Bag	IA-AX	5/21/2013 12:00:00 AM	4/14/2013 12:00:00 AM	4	Black Delete Expired
1	Plastic Bag	AIA	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	4	Red
1	Plastic Bag	AIA	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	4	Red
1	Plastic Bag	EIA-MTX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	2	Red
1	Plastic Bag	EIA-MTX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	10	Red
2	Plastic Bag	EIA-MX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	5	Red
3	Plastic Bag	EIA-MX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM	5	Red
1	Plastic Bag	EIA-TX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM	6	Red

Εικόνα 5.54. Expired.

Παρατηρούμε ότι δίπλα από τους περιέκτες που έχουν λήξει έχει δημιουργηθεί ένας σύνδεσμος *Delete Expired*, και αν ο χρήστης τον ακολουθήσει οι συγκεκριμένοι περιέκτες διαγράφονται από την βάση δεδομένων, και το menu item notification εξαφανίζεται.

Medical Recycling Hello, mike! Log off

Home About Contact

Containers

[Show all containers](#)
Show, edit, delete waste containers

[Submit new containers](#)

[Show waste Categories](#)
Show, edit, delete Categories

[Submit a new Category](#)

[Show clinical waste](#)
Show all occupied/disposed containers

Readers

[Show all Readers](#)
Show, edit, delete Readers

[Submit a new Reader](#)

[Show reader Recordings](#)
Filtered by Reader and/or Bucket

Group by: RFID tag | Category | Date Arrived

Count	Type	Color	Size	Category	DateArrived	DateExpires
20	Hospital box	-	5	EIA-MTX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM
100	Plastic Bag	Black	3	IA-AX	5/16/2013 12:00:00 AM	12/12/2015 12:00:00 AM
1	Plastic Bag	Red	4	AIA	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
9	Plastic Bag	Red	11	AIA	3/18/2013 12:00:00 AM	12/12/2014 12:00:00 AM
1	Plastic Bag	Red	7	AIA	3/18/2013 12:00:00 AM	12/22/2014 12:00:00 AM
1	Plastic Bag	Red	4	AIA	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
1	Plastic Bag	Red	2	EIA-MTX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
1	Plastic Bag	Red	10	EIA-MTX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
2	Plastic Bag	Red	5	EIA-MX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
3	Plastic Bag	Red	5	EIA-MX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
1	Plastic Bag	Red	6	EIA-TX	3/6/2013 12:00:00 AM	1/1/2014 12:00:00 AM
1	Plastic Bag	Red	6	EIA-TX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
2	Plastic Bag	Red	3	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
3	Plastic Bag	Red	5	IA-AX	3/6/2013 12:00:00 AM	11/11/2015 12:00:00 AM
5	Plastic Bag	Yellow	4	AIA	4/11/2013 12:00:00 AM	12/12/2015 12:00:00 AM
11	Plastic Bag	Yellow	11	EIA-MTX	4/4/2013 12:00:00 AM	5/12/2015 12:00:00 AM
2	Plastic Bag	Yellow	3	IA-AX	5/14/2013 12:00:00 AM	4/4/2014 12:00:00 AM

Εικόνα 5.55. Notification out.

Εντελώς αντίστοιχος είναι ο μηχανισμός για τις άλλες ειδοποιήσεις. Πιο συγκεκριμένα όταν εμφανίζεται notification για κάποιον περιέκτη ο οποίος είναι για παραπάνω χρόνο από τον επιτρεπτό μέσα σε έναν συλλέκτη, εμφανίζεται notification, και ο χρήστης καλείται να εκκενώσει τον συγκεκριμένο συλλέκτη σε έναν χώρο προσωρινής αποθήκευσης. Μόλις συμβεί αυτό το notification εξαφανίζεται, όπως είναι αναμενόμενο. Σχετικά με τους χώρους προσωρινής αποθήκευσης, αν η θερμοκρασία είναι μικρότερη από το κατώφλι που έχει ρυθμιστεί στο SQL job, εμφανίζεται κατάλληλη ενημέρωση στον χρήστη, και η ενημέρωση αυτή εξαφανίζεται αν επανέλθει η θερμοκρασία στον χώρο αυτό στα επιτρεπτά επίπεδα.

6 ΕΠΙΛΟΓΟΣ

6.1 Συμπεράσματα

Παρόλο που έχει γίνει αντιληπτό ότι η σωστή και αποτελεσματική διαχείριση των νοσοκομειακών αποβλήτων είναι υψίστης σημασίας τόσο για την προστασία των εργαζομένων και γενικότερα της δημόσιας υγείας, όσο και για την προστασία του περιβάλλοντος, το πλαίσιο δράσης και οι κανόνες που έχουν θεσμοθετηθεί πολύ σπάνια εφαρμόζονται από τις νοσοκομειακές μονάδες ανά την Ελλάδα. Οι κυριότεροι λόγοι που συμβαίνει αυτό είναι η έλλειψη υποδομών αλλά και η έλλειψη εκπαίδευσης και ενημέρωσης των εργαζομένων.

Το πληροφοριακό σύστημα “Medical Recycling” όπως παρουσιάστηκε σε αυτήν την εργασία, συνιστά μια εναλλακτική πρόταση, η οποία θα μπορούσε κάλλιστα να βρει εφαρμογή σε νοσοκομειακές μονάδες, διευκολύνοντας σημαντικά τις διαδικασίες διαχείρισης ιατρικών αποβλήτων και διασφαλίζοντας την τήρηση των θεσμοθετημένων κανόνων και προδιαγραφών που ισχύουν, υπό την προϋπόθεση όμως ότι θα αναπτυχθούν οι απαραίτητες υποδομές και το κατάλληλο οργανωτικό πλαίσιο όπως περιγράφηκαν στα κεφάλαια που προηγήθηκαν. Η εργασία ανάδειξε τη μεγάλη χρηστικότητα της τεχνολογίας ταυτοποίησης μέσω ραδιοσυχνοτήτων καθώς μέσω αυτής αποκτάται σαφής εικόνα σε πραγματικό χρόνο για την κατάσταση των νοσοκομειακών αποβλήτων της νοσοκομειακής μονάδας ενώ παράλληλα παρέχεται συνεχής και έγκαιρη ενημέρωση για πιθανές παραβιάσεις των κανόνων που ισχύουν, καθιστώντας έτσι δυνατή την άμεση ανατροφοδότηση του συστήματος, προκειμένου να γίνουν οι απαραίτητες διορθωτικές ενέργειες και επεμβάσεις αποκατάστασης της ισορροπίας του συστήματος.

Εν κατακλείδι, η παρούσα εργασία θα μπορούσε να αποτελέσει ένα πρότυπο και χρηστικό εργαλείο για τη διαχείριση των νοσοκομειακών αποβλήτων, ειδικά στη χώρα μας όπου ο αντίστοιχος χώρος παρουσιάζει σοβαρά προβλήματα και ελλείψεις με πολύ μεγάλο κόστος για τη δημόσια υγεία.

6.2 Μελλοντικές Επεκτάσεις

Στο παρών πληροφοριακό σύστημα υπήρξε πρόβλεψη έτσι ώστε να εμφανίζονται τα ακριβή στοιχεία τοποθεσίας (συντεταγμένες στον χάρτη) των οχημάτων, έτσι ώστε να διασφαλίζεται η σωστή μεταφορά των απορριμμάτων στην τοποθεσία που προβλέπεται.

Παρ όλα αυτά η υποστήριξη για geolocation είναι ένα σχετικά καινούριο feature της HTML 5, αναπτύσσεται συνεχώς, και δεν έχει ωριμάσει έτσι ώστε να προτυποποιηθεί και να υποστηρίζεται από όλους τους browsers.

Μια πιθανή μελλοντική επέκταση λοιπόν θα ήταν να υπάρχει η δυνατότητα προβολής σε χάρτη της τοποθεσίας των οχημάτων μεταφοράς, και η τοποθεσία αυτή να ανανεώνεται δυναμικά όσο κινείται το όχημα.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Τοποθεσία Web – Ελληνική Εταιρία Κλινικής Χημείας
http://www.eekx-kb.gr/asfaleia_karababa%5B1%5D.pdf
- [2] ΚΥΑ 37591/2031/2003 «Μέτρα και όροι για τη διαχείριση ιατρικών αποβλήτων από υγειονομικές μονάδες» (Β' 1419)
- [3] Θεμιστοκλής Κ. Καραμούστος. Η διαχείριση των επικίνδυνων ιατρικών αποβλήτων στο «Βοστάνειο» Γενικό Νοσοκομείο Μυτιλήνης. Διατριβή στο μεταπτυχιακό πρόγραμμα σπουδών «Περιβαλλοντική πολιτική και διαχείριση του τμήματος περιβάλλοντος», Μυτιλήνη, Νοέμβριος 2005
- [4] **Management Information Systems.** Kenneth Laudon, Jane Laudon, Prentice Hall – International Edition (eighth - 2004)
- [5] Τοποθεσία Web – Microsoft
<http://support.microsoft.com/kb/829019/el>
- [6] Τοποθεσία Web – Asp.NET MVC
<http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>
- [7] Τοποθεσία Web – Csharp Station
<http://www.csharp-station.com/Tutorial.aspx>
- [8] Τοποθεσία Web – Wikipedia
https://en.wikipedia.org/wiki/Ajax_%28programming%29
- [9] Τοποθεσία Web – W3C
http://www.w3schools.com/w3c/w3c_intro.asp
- [10] Τοποθεσία Web – World Wide Web Consortium
<http://www.w3.org/Consortium>
- [11] Τοποθεσία Web – Berkley University
<http://courses.ischool.berkeley.edu/i202/f97/Lecture8/sld040.htm>
- [12] Τοποθεσία Web – Microsoft
<http://msdn.microsoft.com/en-us/library/ms189237.aspx>

- [13] **Συστήματα Βάσεων Δεδομένων**. Silberschatz, Korth, Sudarshan – Εκδόσεις Κλειδάριθμος 2009
- [14] **Inside C#**. Tom Archer, - Εκδόσεις Ελευθερουδάκης 2001
- [15] Τοποθεσία Web – IIS Server
<http://www.iis.net>
- [16] Τοποθεσία Web- RFID Journal
<http://www.rfidjournal.com/>
- [17] Τοποθεσία Web- TI
<http://www.ti.com/rfid/>
- [18] Τοποθεσία Web- IMEC Technologies
<http://www.imectechologies.com/solutions/waste-tracking>
- [19] Smart Trash: Study on RFID tags and the recycling industry. Working paper, interim report (D3) Smart 2010/0042, RAND Europe
- [20] Τοποθεσία Web- Microsoft
<http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx>
- [21] Τοποθεσία Web- Microsoft
<http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx>
- [22] Τοποθεσία Web- W3C
http://www.w3schools.com/html/html5_intro.asp
- [23] Τοποθεσία Web- W3C
<http://www.w3schools.com/css3/>
- [24] Τοποθεσία Web- JQuery
<http://learn.jquery.com/>
- [25] Τοποθεσία Web- JQuery Cycle
<http://jquery.malsup.com/cycle/>
- [26] Τοποθεσία Web- W3C
<http://www.w3schools.com/js/>
- [27] Τοποθεσία Web- W3C
<http://www.w3schools.com/ajax/>

[28] Ηλίας Σκούμας. Διαχείριση ιατρικών αποβλήτων. Διπλωματική εργασία, Μάρτιος 2013, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών, Αθήνα.

8 ΠΑΡΑΡΤΗΜΑ

Λόγω της αρκετά μεγάλης ποσότητας του κώδικα που υλοποιεί το συγκεκριμένο πληροφοριακό σύστημα, και του γεγονότος ότι μπορεί κάποιος να κατεβάσει τον κώδικα HTML, CSS, JavaScript μιας ιστοσελίδας από τον browser που χρησιμοποιεί, κρίθηκε σκόπιμο να παραλειφθεί ένα μέρος του κώδικα. Το σημαντικότερο μέρος της εφαρμογής, είναι η βάση δεδομένων, αφού υλοποιεί όλα τα business rules και κρατάει όλα τα απαραίτητα δεδομένα. Γι αυτόν τον σκοπό παραθέτουμε στο παράρτημα μόνο το ddl script που δημιουργεί την βάση αυτή.

Το script που δημιουργεί την βάση δεδομένων :

```
USE [master]
GO
/***** Object: Database [WasteManagement] Script Date: 05/21/2013 23:33:00 *****/
CREATE DATABASE [WasteManagement] ON PRIMARY
( NAME = N'WasteManagement', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\WasteManagement.mdf' , SIZE =
3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'WasteManagement_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\WasteManagement_log.ldf' , SIZE =
1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [WasteManagement] SET COMPATIBILITY_LEVEL = 100
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [WasteManagement].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [WasteManagement] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [WasteManagement] SET ANSI_NULLS OFF
GO
ALTER DATABASE [WasteManagement] SET ANSI_PADDING OFF
GO
ALTER DATABASE [WasteManagement] SET ANSI_WARNINGS OFF
GO
```



```

ALTER DATABASE [WasteManagement] SET ARITHABORT OFF
GO
ALTER DATABASE [WasteManagement] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [WasteManagement] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [WasteManagement] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [WasteManagement] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [WasteManagement] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [WasteManagement] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [WasteManagement] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [WasteManagement] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [WasteManagement] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [WasteManagement] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [WasteManagement] SET DISABLE_BROKER
GO
ALTER DATABASE [WasteManagement] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [WasteManagement] SET DATE_CORRELATION_OPTIMIZATION
OFF
GO
ALTER DATABASE [WasteManagement] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [WasteManagement] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [WasteManagement] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [WasteManagement] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [WasteManagement] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [WasteManagement] SET READ_WRITE
GO
ALTER DATABASE [WasteManagement] SET RECOVERY FULL
GO
ALTER DATABASE [WasteManagement] SET MULTI_USER
GO
ALTER DATABASE [WasteManagement] SET PAGE_VERIFY CHECKSUM

```

```

GO
ALTER DATABASE [WasteManagement] SET DB_CHAINING OFF
GO
USE [WasteManagement]
GO
/***** Object: Table [dbo].[Bucket]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Bucket](
    [BucketID] [int] IDENTITY(1,1) NOT NULL,
    [RFIDtag] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Location] [nvarchar](50) NOT NULL,
    [Capacity] [decimal](10, 1) NOT NULL,
    [CurrentWeight] [decimal](10, 1) NOT NULL,
    [CurrentTemperature] [decimal](10, 1) NOT NULL,
    [IsEmpty] [bit] NOT NULL,
    [Description] [nvarchar](50) NULL,
    CONSTRAINT [PK_Bin] PRIMARY KEY CLUSTERED
(
    [BucketID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Fridge]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Fridge](
    [FridgeID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Location] [nvarchar](50) NOT NULL,
    [Capacity] [decimal](10, 1) NOT NULL,
    [CurrentWeight] [decimal](10, 1) NOT NULL,
    [CurrentTemperature] [decimal](10, 1) NOT NULL,
    [Description] [nvarchar](100) NULL,
    CONSTRAINT [PK_Fridge] PRIMARY KEY CLUSTERED
(
    [FridgeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Driver]   Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Driver](
    [DriverID] [int] IDENTITY(1,1) NOT NULL,
    [FirstName] [nvarchar](20) NOT NULL,
    [LastName] [nvarchar](20) NOT NULL,
    [PhoneNumber] [nvarchar](25) NOT NULL,
    [Address] [nvarchar](50) NOT NULL,
    [EmailAdress] [nvarchar](30) NULL,
    [OtherNotes] [nvarchar](100) NULL,
    CONSTRAINT [PK_Driver] PRIMARY KEY CLUSTERED
(
    [DriverID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ClinicalWaste]   Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ClinicalWaste](
    [ClinicalWaste] [int] IDENTITY(1,1) NOT NULL,
    [CategoryID] [nvarchar](50) NOT NULL,
    [RFIDtag] [uniqueidentifier] NOT NULL,
    [DateCreated] [datetime2](7) NOT NULL,
    [DateDestroyed] [datetime2](7) NULL,
    [BucketID] [nvarchar](50) NOT NULL,
    [Weight] [decimal](10, 1) NOT NULL,
    [DateExpires] [datetime2](7) NULL,
    [InFridge] [bit] NOT NULL,
    [FridgeID] [nvarchar](50) NULL,
    CONSTRAINT [PK_ClinicalWaste] PRIMARY KEY CLUSTERED
(
    [ClinicalWaste] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Category]   Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Category](
    [CategoryID] [int] IDENTITY(1,1) NOT NULL,
    [CategoryName] [nvarchar](50) NOT NULL,
    [Description] [nvarchar](200) NULL,
    [OutsideTheRefrigrator] [decimal](10, 1) NOT NULL,
    CONSTRAINT [PK_Category] PRIMARY KEY CLUSTERED
(
    [CategoryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [IX_Category] UNIQUE NONCLUSTERED
(
    [CategoryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Truck]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Truck](
    [TruckID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](50) NOT NULL,
    [Capacity] [decimal](10, 1) NOT NULL,
    [CurrentWeight] [decimal](10, 1) NULL,
    [CurrentTemperature] [decimal](10, 1) NULL,
    [CurrentLocation] [geography] NULL,
    CONSTRAINT [PK_Truck] PRIMARY KEY CLUSTERED
(
    [TruckID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Reader]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Reader](
    [ReaderID] [int] IDENTITY(1,1) NOT NULL,

```

```

        [Name] [nvarchar](50) NOT NULL,
        [Location] [nvarchar](50) NOT NULL,
        [Description] [nvarchar](100) NULL,
    CONSTRAINT [PK_Reader] PRIMARY KEY CLUSTERED
    (
        [ReaderID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
    = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Notifications]   Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Notifications](
    [NotificationId] [int] IDENTITY(1,1) NOT NULL,
    [type] [nvarchar](50) NOT NULL,
    [info] [nvarchar](200) NOT NULL,
    CONSTRAINT [PK_Notifications] PRIMARY KEY CLUSTERED
    (
        [NotificationId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
    = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[InsertTrucks]   Script Date: 05/21/2013 23:33:01
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertTrucks]
    -- Add the parameters for the stored procedure here
    @name nvarchar(50),
    @capacity decimal(10,1)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from

```

```

-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
INSERT INTO Truck
(Name,Capacity)
VALUES
(@name,@capacity)

END
GO
/***** Object: View [dbo].[vw_BucketTruck]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BucketTruck] AS
SELECT Truck.Name AS Truck, Bucket.Name as Bucket,BucketTruck.DateAndTime AS
DateAndTime, BucketTruck.Location as Location
FROM BucketTruck
JOIN Truck
ON BucketTruck.TruckID = Truck.TruckID
JOIN Bucket
ON Bucket.BucketID = BucketTruck.BucketID
GO
/***** Object: Table [dbo].[BucketReader]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BucketReader](
    [BucketID] [int] NOT NULL,
    [ReaderID] [int] NOT NULL,
    [DateAndTime] [datetime2](7) NOT NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Bag]  Script Date: 05/21/2013 23:33:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Bag](
    [BagID] [int] IDENTITY(1,1) NOT FOR REPLICATION NOT NULL,
    [RFIDtag] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [DateArrived] [date] NOT NULL,
    [DateExpires] [date] NOT NULL,

```

```

[Size] [int] NOT NULL,
[IsEmpty] [bit] NOT NULL,
[CategoryID] [int] NOT NULL,
[IsDisposed] [bit] NOT NULL,
[Color] [nvarchar](50) NOT NULL,
[Type] [nvarchar](50) NOT NULL,
[OtherInfo] [nvarchar](50) NULL,
CONSTRAINT [PK_Bag] PRIMARY KEY CLUSTERED
(
    [BagID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[InsertFridge]  Script Date: 05/21/2013 23:33:01
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertFridge]
    -- Add the parameters for the stored procedure here
    @name nvarchar(50),
    @location nvarchar(50),
    @capacity decimal(10,1),
    @description nvarchar(50)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO Fridge
    (Name,Location,Capacity,Description)
    VALUES
    (@name,@location,@capacity,@description)

END

```

```

GO
/***** Object: StoredProcedure [dbo].[InsertBuckets]  Script Date: 05/21/2013 23:33:01
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertBuckets]
    -- Add the parameters for the stored procedure here
    @name nvarchar(50),
    @location nvarchar(50),
    @capacity decimal(10,1),
    @Description nvarchar(50)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO Bucket
    (Name,Location,Capacity,Description)
    VALUES
    (@name,@location,@capacity,@Description)

END
GO
/***** Object: Table [dbo].[BagBucket]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BagBucket](
    [BagID] [int] NOT NULL,
    [BucketID] [int] NOT NULL,
    [DateAndTime] [datetime] NOT NULL,
    [BagWeight] [decimal](10, 1) NOT NULL
) ON [PRIMARY]
GO

```



```
/****** Object: StoredProcedure [dbo].[InsertBucketReader]  Script Date: 05/21/2013
23:33:01 *****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
```

```
CREATE PROCEDURE [dbo].[InsertBucketReader]
    -- Add the parameters for the stored procedure here
    @bucket int,
    @reader int
```

```
AS
BEGIN
```

```
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
```

```
    -- Insert statements for procedure here
    INSERT INTO BucketReader
    (BucketID,ReaderID)
    VALUES
    (@bucket,@reader)
```

```
END
GO
```

```
/****** Object: StoredProcedure [dbo].[insertBags]  Script Date: 05/21/2013 23:33:01
*****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
-- =====
-- Author:          Mike Milonakis
-- Create date:
-- Description: submission of new bags
-- =====
```

```
CREATE PROCEDURE [dbo].[insertBags]
    -- Add the parameters for the stored procedure here
    @quantity int = 1,
    @expdate date ,
```

```

        @size int,
        @category int ,
        @color nvarchar(50),
        @type nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    DECLARE @count int = 0;

    WHILE @count < @quantity
    BEGIN
        IF (@color IS NULL)
        BEGIN
            SET @color = '-'
        END

        INSERT INTO dbo.Bag (DateExpires,Size,CategoryID,Color,Type)
        VALUES (@expdate,@size,@category,@color,@type);
        SET @count = @count + 1
    END
END
GO
/***** Object: View [dbo].[vw_BucketReader]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BucketReader] AS
SELECT Reader.Location AS Reader, Bucket.Name AS Bucket, BucketReader.DateAndTime
AS DateAndTime
FROM BucketReader
JOIN Bucket
ON BucketReader.BucketID = Bucket.BucketID
JOIN Reader
ON BucketReader.ReaderID = Reader.ReaderID
GO
/***** Object: View [dbo].[vw_BagsBySize]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BagsBySize] AS

```

```

SELECT COUNT(BagID) AS Count, Category.CategoryName AS Category, Bag.DateArrived
AS DateArrived, Bag.DateExpires as DateExpires, Bag.Size AS Size
FROM Bag
JOIN Category
ON Bag.CategoryID = Category.CategoryID
GROUP BY Size,CategoryName,DateArrived,DateExpires
GO
/***** Object: View [dbo].[vw_BagsByCategory]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BagsByCategory]
AS
SELECT  dbo.Bag.Type, COUNT(dbo.Bag.BagID) AS Count, dbo.Bag.Color,
dbo.Category.CategoryName AS Category, dbo.Bag.DateArrived, dbo.Bag.DateExpires,
        dbo.Bag.Size
FROM    dbo.Bag INNER JOIN
        dbo.Category ON dbo.Bag.CategoryID = dbo.Category.CategoryID
GROUP BY dbo.Bag.Type, dbo.Bag.Color, dbo.Category.CategoryName, dbo.Bag.DateExpires,
dbo.Bag.DateArrived, dbo.Bag.Size
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-
B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
    Begin PaneConfigurations =
        Begin PaneConfiguration = 0
            NumPanes = 4
            Configuration = "(H (1[40] 4[20] 2[20] 3) )"
        End
        Begin PaneConfiguration = 1
            NumPanes = 3
            Configuration = "(H (1 [50] 4 [25] 3))"
        End
        Begin PaneConfiguration = 2
            NumPanes = 3
            Configuration = "(H (1 [50] 2 [25] 3))"
        End
        Begin PaneConfiguration = 3
            NumPanes = 3
            Configuration = "(H (4 [30] 2 [40] 3))"
        End
        Begin PaneConfiguration = 4
            NumPanes = 2
            Configuration = "(H (1 [56] 3))"
        End
    End

```

```

Begin PaneConfiguration = 5
  NumPanels = 2
  Configuration = "(H (2 [66] 3))"
End
Begin PaneConfiguration = 6
  NumPanels = 2
  Configuration = "(H (4 [50] 3))"
End
Begin PaneConfiguration = 7
  NumPanels = 1
  Configuration = "(V (3))"
End
Begin PaneConfiguration = 8
  NumPanels = 3
  Configuration = "(H (1[56] 4[18] 2) )"
End
Begin PaneConfiguration = 9
  NumPanels = 2
  Configuration = "(H (1 [75] 4))"
End
Begin PaneConfiguration = 10
  NumPanels = 2
  Configuration = "(H (1[66] 2) )"
End
Begin PaneConfiguration = 11
  NumPanels = 2
  Configuration = "(H (4 [60] 2))"
End
Begin PaneConfiguration = 12
  NumPanels = 1
  Configuration = "(H (1) )"
End
Begin PaneConfiguration = 13
  NumPanels = 1
  Configuration = "(V (4))"
End
Begin PaneConfiguration = 14
  NumPanels = 1
  Configuration = "(V (2))"
End
ActivePaneConfig = 0
End
Begin DiagramPane =
  Begin Origin =
    Top = 0
    Left = 0

```

```

End
Begin Tables =
  Begin Table = "Bag"
    Begin Extent =
      Top = 6
      Left = 38
      Bottom = 125
      Right = 214
    End
    DisplayFlags = 280
    TopColumn = 0
  End
  Begin Table = "Category"
    Begin Extent =
      Top = 6
      Left = 252
      Bottom = 125
      Right = 467
    End
    DisplayFlags = 280
    TopColumn = 0
  End
End
End
End
Begin SQLPane =
End
Begin DataPane =
  Begin ParameterDefaults = ""
  End
  Begin ColumnWidths = 9
    Width = 284
    Width = 1500
    Width = 1500
    Width = 1500
    Width = 1500
    Width = 1500
    Width = 1500
    Width = 1500
    Width = 1500
  End
End
End
Begin CriteriaPane =
  Begin ColumnWidths = 12
    Column = 1440
    Alias = 900
    Table = 1170

```

```

Output = 720
Append = 1400
NewValue = 1170
SortType = 1350
SortOrder = 1410
GroupBy = 1350
Filter = 1350
Or = 1350
Or = 1350
Or = 1350
End
End
End
', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vw_BagsByCategory'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vw_BagsByCategory'
GO
/***** Object: View [dbo].[vw_BagsBy_Date_Arrived]  Script Date: 05/21/2013 23:33:01
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BagsBy_Date_Arrived]
AS
SELECT  dbo.Bag.Type, COUNT(dbo.Bag.BagID) AS Count, dbo.Bag.Color,
dbo.Category.CategoryName AS CATEGORY, dbo.Bag.DateArrived, dbo.Bag.DateExpires,
        dbo.Bag.Size
FROM    dbo.Bag INNER JOIN
        dbo.Category ON dbo.Bag.CategoryID = dbo.Category.CategoryID
GROUP BY dbo.Bag.Type, dbo.Bag.Color, dbo.Bag.DateArrived, dbo.Category.CategoryName,
dbo.Bag.Size, dbo.Bag.DateExpires
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-
B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
    Begin PaneConfigurations =
        Begin PaneConfiguration = 0
            NumPanes = 4
            Configuration = "(H (1[40] 4[20] 2[20] 3) )"
        End
        Begin PaneConfiguration = 1
            NumPanes = 3

```

```

    Configuration = "(H (1 [50] 4 [25] 3))"
End
Begin PaneConfiguration = 2
    NumPanels = 3
    Configuration = "(H (1 [50] 2 [25] 3))"
End
Begin PaneConfiguration = 3
    NumPanels = 3
    Configuration = "(H (4 [30] 2 [40] 3))"
End
Begin PaneConfiguration = 4
    NumPanels = 2
    Configuration = "(H (1 [56] 3))"
End
Begin PaneConfiguration = 5
    NumPanels = 2
    Configuration = "(H (2 [66] 3))"
End
Begin PaneConfiguration = 6
    NumPanels = 2
    Configuration = "(H (4 [50] 3))"
End
Begin PaneConfiguration = 7
    NumPanels = 1
    Configuration = "(V (3))"
End
Begin PaneConfiguration = 8
    NumPanels = 3
    Configuration = "(H (1[56] 4[18] 2) )"
End
Begin PaneConfiguration = 9
    NumPanels = 2
    Configuration = "(H (1 [75] 4))"
End
Begin PaneConfiguration = 10
    NumPanels = 2
    Configuration = "(H (1[66] 2) )"
End
Begin PaneConfiguration = 11
    NumPanels = 2
    Configuration = "(H (4 [60] 2))"
End
Begin PaneConfiguration = 12
    NumPanels = 1
    Configuration = "(H (1) )"
End

```

```

Begin PaneConfiguration = 13
  NumPanes = 1
  Configuration = "(V (4))"
End
Begin PaneConfiguration = 14
  NumPanes = 1
  Configuration = "(V (2))"
End
ActivePaneConfig = 0
End
Begin DiagramPane =
  Begin Origin =
    Top = 0
    Left = 0
  End
  Begin Tables =
    Begin Table = "Bag"
      Begin Extent =
        Top = 6
        Left = 38
        Bottom = 125
        Right = 214
      End
      DisplayFlags = 280
      TopColumn = 0
    End
    Begin Table = "Category"
      Begin Extent =
        Top = 6
        Left = 252
        Bottom = 125
        Right = 467
      End
      DisplayFlags = 280
      TopColumn = 0
    End
  End
End
Begin SQLPane =
End
Begin DataPane =
  Begin ParameterDefaults = ""
  End
End
Begin CriteriaPane =
  Begin ColumnWidths = 12

```



```

Column = 1440
Alias = 900
Table = 1170
Output = 720
Append = 1400
NewValue = 1170
SortType = 1350
SortOrder = 1410
GroupBy = 1350
Filter = 1350
Or = 1350
Or = 1350
Or = 1350
End
End
End
', @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vw_BagsBy_Date_Arrived'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vw_BagsBy_Date_Arrived'
GO
/***** Object: Trigger [trig_WasteGeneration] Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_WasteGeneration]
ON [dbo].[BagBucket]
AFTER INSERT
AS

BEGIN

    DECLARE @bag AS INT
    DECLARE @weight AS decimal(10, 1)
    DECLARE @time AS DATETIME2
    DECLARE @bucket AS INT
    DECLARE @tag AS UNIQUEIDENTIFIER
    DECLARE @category AS INT
    DECLARE @exptime AS DECIMAL(10,1)
    DECLARE @bucketname AS NVARCHAR(50)
    DECLARE @categoryname AS NVARCHAR(50)

    SELECT @bag = BagID

```

```

FROM inserted

SELECT @bucket = BucketID
FROM inserted

SELECT @weight = BagWeight
FROM inserted

SELECT @time = DateAndTime
FROM inserted

SELECT @tag = RFIDtag
FROM Bag
WHERE Bag.BagID = @bag

SELECT @category = Bag.CategoryID
FROM Bag
WHERE Bag.BagID = @bag

SELECT @exptime = OutsideTheRefrigorator
FROM Category
WHERE CategoryID = @category

SELECT @categoryname = Category.CategoryName
FROM Category
WHERE CategoryID = @category

SELECT @bucketname = Bucket.Name
FROM Bucket
WHERE BucketID = @bucket

INSERT INTO ClinicalWaste
(RFIDtag,DateCreated,BucketID,weight,CategoryID,DateExpires)
VALUES
(@tag,@time,@bucketname,@weight,@categoryname,DATEADD(HOUR,@exptime,@
time))

END
GO
/***** Object: Trigger [trig_AddWeight]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_AddWeight]
ON [dbo].[BagBucket]

```

```

AFTER INSERT
AS
BEGIN

    DECLARE @weight AS decimal(10, 1)
    DECLARE @bucket AS int

    SELECT @bucket = BucketID
    FROM inserted

    SELECT @weight = BagWeight
    FROM inserted

    UPDATE Bucket
    SET CurrentWeight = CurrentWeight + @weight, IsEmpty = 0
    WHERE BucketID = @bucket
END
GO
/***** Object: View [dbo].[vw_BagBucket]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vw_BagBucket] AS
SELECT TOP 99 PERCENT Bag.RFIDtag AS Bag, Bucket.Name AS Bucket,
BagBucket.DateAndTime AS DateAndTime
FROM BagBucket
JOIN Bucket
ON BagBucket.BucketID = Bucket.BucketID
JOIN Bag
ON BagBucket.BagID = Bag.BagID
ORDER BY BagBucket.DateAndTime DESC
GO
/***** Object: Trigger [trig_BagPopulation]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_BagPopulation]
ON [dbo].[BagBucket]
AFTER INSERT
AS
BEGIN
    DECLARE @bag AS INT

    SELECT @bag = BagID

```

```

        FROM inserted

        UPDATE Bag
        SET IsEmpty = 0
        WHERE BagID = @bag
END
GO
/***** Object: Table [dbo].[BucketFridge]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BucketFridge](
    [BucketID] [int] NOT NULL,
    [FridgeID] [int] NOT NULL,
    [DateAndTime] [datetime] NOT NULL
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[InsertBagBucket]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertBagBucket]
    -- Add the parameters for the stored procedure here
    @bucket int,
    @bag int,
    @bagweight decimal(10,1)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO BagBucket
    (BucketID,BagID,BagWeight)
    VALUES

```

```

        (@bucket,@bag,@bagweight)

END
GO
/***** Object: StoredProcedure [dbo].[InsertBucketFridge]  Script Date: 05/21/2013
23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertBucketFridge]
    -- Add the parameters for the stored procedure here
    @bucket int,
    @fridge int

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO BucketFridge
    (BucketID,FridgeID)
    VALUES
    (@bucket,@fridge)

END
GO
/***** Object: Table [dbo].[FridgeTruck]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FridgeTruck](
    [FridgeID] [int] NOT NULL,
    [TruckID] [int] NOT NULL,
    [RouteID] [int] NULL,
    [Location] [geography] NULL,
    [DateAndTime] [datetime2](7) NOT NULL

```

```

) ON [PRIMARY]
GO
/***** Object: Trigger [trig_EmptyBucket_WasteinFridge]  Script Date: 05/21/2013
23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_EmptyBucket_WasteinFridge]
ON [dbo].[BucketFridge]
AFTER INSERT
AS
BEGIN

    DECLARE @weight AS decimal(10, 1)
    DECLARE @bucket AS int
    DECLARE @fridge AS int
    DECLARE @fridgeName AS NVARCHAR(50)
    DECLARE @bucketName AS NVARCHAR(50)

    SELECT @bucket = BucketID
    FROM inserted

    SELECT @bucketName = Bucket.Name
    FROM Bucket
    WHERE BucketID = @bucket

    SELECT @fridge = FridgeID
    FROM inserted

    SELECT @fridgeName = Fridge.Name
    FROM Fridge
    WHERE FridgeID = @fridge

    SELECT @weight = Bucket.CurrentWeight
    FROM Bucket
    WHERE Bucket.BucketID = @bucket

    BEGIN

        UPDATE Fridge
        SET CurrentWeight = CurrentWeight + @weight
        WHERE Fridge.FridgeID = @fridge

    --Emptybucketswasteinfridge

```

```
UPDATE ClinicalWaste
SET InFridge = 1, FridgeID = @fridgeName
WHERE (BucketID = @bucketName AND FridgeID is NULL)
```

```
--TruncBucketWeight
UPDATE Bucket
SET CurrentWeight = 0 , IsEmpty = 1
WHERE BucketID = @bucket
```

```
--EmptyBucket
DELETE FROM BagBucket
WHERE BagBucket.BucketID = @bucket
```

```
END;
```

```
END
```

```
GO
```

```
/***** Object: Table [dbo].[Route]  Script Date: 05/21/2013 23:33:01 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Route](
```

```
    [RouteID] [int] IDENTITY(1,1) NOT NULL,
```

```
    [DriverID] [int] NOT NULL,
```

```
    [TruckID] [int] NOT NULL,
```

```
    [DateAndTime] [datetime] NULL,
```

```
    [StartingLocation] [geography] NULL,
```

```
    [Destination] [geography] NULL,
```

```
    [IsCompleted] [bit] NOT NULL,
```

```
CONSTRAINT [PK_Route] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [RouteID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
```

```
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[InsertFridgeTruck]  Script Date: 05/21/2013 23:33:01
```

```
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

GO

```
-- =====  
-- Author:          <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====
```

```
CREATE PROCEDURE [dbo].[InsertFridgeTruck]  
    -- Add the parameters for the stored procedure here  
    @truck int,  
    @fridge int
```

AS

BEGIN

```
-- SET NOCOUNT ON added to prevent extra result sets from  
-- interfering with SELECT statements.  
SET NOCOUNT ON;
```

```
-- Insert statements for procedure here  
INSERT INTO FridgeTruck  
    (TruckID,FridgeID)  
VALUES  
    (@truck,@fridge)
```

END

GO

```
/***** Object: Trigger [trig_AddWeightToTruck]   Script Date: 05/21/2013 23:33:01 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TRIGGER [dbo].[trig_AddWeightToTruck]
```

```
ON [dbo].[FridgeTruck]
```

```
AFTER INSERT
```

AS

BEGIN

```
    DECLARE @weight AS decimal(10, 1)  
    DECLARE @truck AS int  
    DECLARE @fridge AS int  
    DECLARE @date AS datetime2  
    DECLARE @fridgeName AS NVARCHAR(50)
```

```
    SELECT @date = DateAndTime  
    FROM inserted
```



```

SELECT @truck = TruckID
FROM inserted

SELECT @fridge = FridgeID
FROM inserted

SELECT @fridgeName = Fridge.Name
FROM Fridge
WHERE FridgeID = @fridge

SELECT @weight = Fridge.CurrentWeight
FROM Fridge
WHERE Fridge.FridgeID = @fridge

--AddWeightToTruck
UPDATE Truck
SET CurrentWeight = CurrentWeight + @weight
WHERE Truck.TruckID = @truck

--EmptyFridgeDateDisposed
UPDATE ClinicalWaste
SET DateDestroyed = @date
WHERE ((DateDestroyed is NULL) AND (FridgeID = @fridgeName))

--ClinicalWasteBagDisposed
UPDATE Bag
SET IsDisposed = 1
WHERE (Bag.RFIDtag IN
      (SELECT RFIDtag FROM ClinicalWaste
       WHERE (FridgeID = @fridgeName AND DateDestroyed = @date)))

--Emptyfridge
UPDATE Fridge
SET CurrentWeight = 0
WHERE Fridge.FridgeID = @fridge

DELETE FROM BucketFridge
WHERE BucketFridge.FridgeID = @fridge
END
GO
/***** Object: View [dbo].[vw_Route]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE VIEW [dbo].[vw_Route] AS
SELECT Driver.FirstName + ' ' + Driver.LastName AS Driver, Truck.Name AS
Truck,Route.StartingLocation AS StartingLocation, Route.Destination as Destination,
Route.DateAndTime as DateAndTime, Route.IsCompleted as Finished
FROM Route
JOIN Driver
ON Driver.DriverID = Route.DriverID
JOIN Truck
ON Truck.TruckID = Route.TruckID
GO
/***** Object: Trigger [trig_Route]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_Route]
ON [dbo].[Route]
AFTER INSERT
AS
BEGIN
    DECLARE @route AS int
    DECLARE @truck AS int

    SELECT @truck = TruckID
    FROM inserted

    SELECT @route = RouteID
    FROM inserted

    UPDATE FridgeTruck
    SET RouteID = @route
    WHERE ((TruckID = @truck) AND (RouteID IS NULL))
END
GO
/***** Object: Trigger [trig_EmptyTruck]  Script Date: 05/21/2013 23:33:01 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER [dbo].[trig_EmptyTruck]
ON [dbo].[Route]
AFTER UPDATE
AS
BEGIN
    DECLARE @route AS int

```

```

DECLARE @truck AS int
DECLARE @IsCompleted AS BIT

SELECT @truck = TruckID
FROM inserted

SELECT @route = RouteID
FROM inserted

SELECT @IsCompleted = IsCompleted
FROM inserted

IF (@IsCompleted = 1)
BEGIN
    UPDATE Truck
    SET CurrentWeight = 0
    WHERE (TruckID = @truck)
END
END
GO
/***** Object: StoredProcedure [dbo].[InsertRoutes]   Script Date: 05/21/2013 23:33:01
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[InsertRoutes]
    -- Add the parameters for the stored procedure here
    @driver int,
    @truck int

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    INSERT INTO Route
    (DriverID,TruckID)

```

```

VALUES
(@driver,@truck)

END
GO
/***** Object: Default [DF_Truck_CurrentWeight]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[Truck] ADD CONSTRAINT [DF_Truck_CurrentWeight] DEFAULT
('0') FOR [CurrentWeight]
GO
/***** Object: Default [DF_Truck_CurrentTemperature]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[Truck] ADD CONSTRAINT [DF_Truck_CurrentTemperature]
DEFAULT ((25)) FOR [CurrentTemperature]
GO
/***** Object: Default [DF_Bin_RFIDtag]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bucket] ADD CONSTRAINT [DF_Bin_RFIDtag] DEFAULT
(newid()) FOR [RFIDtag]
GO
/***** Object: Default [DF_Bin_CurrentWeight]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bucket] ADD CONSTRAINT [DF_Bin_CurrentWeight] DEFAULT
('0') FOR [CurrentWeight]
GO
/***** Object: Default [DF_Bin_CurrentTemperature]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[Bucket] ADD CONSTRAINT [DF_Bin_CurrentTemperature]
DEFAULT ('25') FOR [CurrentTemperature]
GO
/***** Object: Default [DF_Bucket_IsEmpty]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bucket] ADD CONSTRAINT [DF_Bucket_IsEmpty] DEFAULT ((1))
FOR [IsEmpty]
GO
/***** Object: Default [DF_Fridge_CurrentWeight]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[Fridge] ADD CONSTRAINT [DF_Fridge_CurrentWeight] DEFAULT
((0)) FOR [CurrentWeight]
GO
/***** Object: Default [DF_Fridge_CurrentTemperature]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[Fridge] ADD CONSTRAINT [DF_Fridge_CurrentTemperature]
DEFAULT ((3)) FOR [CurrentTemperature]
GO
/***** Object: Default [DF_Fridge_Description]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Fridge] ADD CONSTRAINT [DF_Fridge_Description] DEFAULT (")
FOR [Description]
GO

```

```

/***** Object: Default [DF_ClinicalWaste_DateCreated]   Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[ClinicalWaste] ADD CONSTRAINT [DF_ClinicalWaste_DateCreated]
DEFAULT (getdate()) FOR [DateCreated]
GO
/***** Object: Default [DF_ClinicalWaste_InFridge]   Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[ClinicalWaste] ADD CONSTRAINT [DF_ClinicalWaste_InFridge]
DEFAULT ((0)) FOR [InFridge]
GO
/***** Object: Default [DF_BinReader_DateAndTime]   Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[BucketReader] ADD CONSTRAINT [DF_BinReader_DateAndTime]
DEFAULT (getdate()) FOR [DateAndTime]
GO
/***** Object: Default [DF_Bag_RFIDtag]   Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bag] ADD CONSTRAINT [DF_Bag_RFIDtag] DEFAULT (newid())
FOR [RFIDtag]
GO
/***** Object: Default [DF_Bag_DateArrived]   Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bag] ADD CONSTRAINT [DF_Bag_DateArrived] DEFAULT
(getdate()) FOR [DateArrived]
GO
/***** Object: Default [DF_Bag_IsEmpty]   Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bag] ADD CONSTRAINT [DF_Bag_IsEmpty] DEFAULT ((1)) FOR
[IsEmpty]
GO
/***** Object: Default [DF_Bag_IsDisposed]   Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bag] ADD CONSTRAINT [DF_Bag_IsDisposed] DEFAULT ((0))
FOR [IsDisposed]
GO
/***** Object: Default [DF_BagBin_DateAndTime]   Script Date: 05/21/2013 23:33:01
*****/
ALTER TABLE [dbo].[BagBucket] ADD CONSTRAINT [DF_BagBin_DateAndTime]
DEFAULT (getdate()) FOR [DateAndTime]
GO
/***** Object: Default [DF_BucketFridge_DateAndTime]   Script Date: 05/21/2013 23:33:01
*****/
ALTER TABLE [dbo].[BucketFridge] ADD CONSTRAINT
[DF_BucketFridge_DateAndTime] DEFAULT (getdate()) FOR [DateAndTime]
GO
/***** Object: Default [DF_FridgeTruck_DateAndTime]   Script Date: 05/21/2013 23:33:01
*****/
ALTER TABLE [dbo].[FridgeTruck] ADD CONSTRAINT [DF_FridgeTruck_DateAndTime]
DEFAULT (getdate()) FOR [DateAndTime]
GO

```

```

/***** Object: Default [DF_Route_DateAndTime]  Script Date: 05/21/2013 23:33:01
*****/
ALTER TABLE [dbo].[Route] ADD CONSTRAINT [DF_Route_DateAndTime] DEFAULT
(getdate()) FOR [DateAndTime]
GO
/***** Object: Default [DF_Route_IsCompleted]  Script Date: 05/21/2013 23:33:01 *****/
ALTER TABLE [dbo].[Route] ADD CONSTRAINT [DF_Route_IsCompleted] DEFAULT
((0)) FOR [IsCompleted]
GO
/***** Object: ForeignKey [FK_BinReader_Bin]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[BucketReader] WITH CHECK ADD CONSTRAINT
[FK_BinReader_Bin] FOREIGN KEY([BucketID])
REFERENCES [dbo].[Bucket] ([BucketID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BucketReader] CHECK CONSTRAINT [FK_BinReader_Bin]
GO
/***** Object: ForeignKey [FK_BinReader_Reader]  Script Date: 05/21/2013 23:33:00
*****/
ALTER TABLE [dbo].[BucketReader] WITH CHECK ADD CONSTRAINT
[FK_BinReader_Reader] FOREIGN KEY([ReaderID])
REFERENCES [dbo].[Reader] ([ReaderID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BucketReader] CHECK CONSTRAINT [FK_BinReader_Reader]
GO
/***** Object: ForeignKey [FK_Bag_Category]  Script Date: 05/21/2013 23:33:00 *****/
ALTER TABLE [dbo].[Bag] WITH CHECK ADD CONSTRAINT [FK_Bag_Category]
FOREIGN KEY([CategoryID])
REFERENCES [dbo].[Category] ([CategoryID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Bag] CHECK CONSTRAINT [FK_Bag_Category]
GO
/***** Object: ForeignKey [FK_BagBucket_Bag]  Script Date: 05/21/2013 23:33:01 *****/
ALTER TABLE [dbo].[BagBucket] WITH CHECK ADD CONSTRAINT
[FK_BagBucket_Bag] FOREIGN KEY([BagID])
REFERENCES [dbo].[Bag] ([BagID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BagBucket] CHECK CONSTRAINT [FK_BagBucket_Bag]
GO

```

/****** Object: ForeignKey [FK_BagBucket_Bucket] Script Date: 05/21/2013 23:33:01
*****/

```
ALTER TABLE [dbo].[BagBucket] WITH CHECK ADD CONSTRAINT
[FK_BagBucket_Bucket] FOREIGN KEY([BucketID])
REFERENCES [dbo].[Bucket] ([BucketID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BagBucket] CHECK CONSTRAINT [FK_BagBucket_Bucket]
GO
```

/****** Object: ForeignKey [FK_BucketFridge_Bucket] Script Date: 05/21/2013 23:33:01
*****/

```
ALTER TABLE [dbo].[BucketFridge] WITH CHECK ADD CONSTRAINT
[FK_BucketFridge_Bucket] FOREIGN KEY([BucketID])
REFERENCES [dbo].[Bucket] ([BucketID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BucketFridge] CHECK CONSTRAINT [FK_BucketFridge_Bucket]
GO
```

/****** Object: ForeignKey [FK_BucketFridge_Fridge] Script Date: 05/21/2013 23:33:01
*****/

```
ALTER TABLE [dbo].[BucketFridge] WITH CHECK ADD CONSTRAINT
[FK_BucketFridge_Fridge] FOREIGN KEY([FridgeID])
REFERENCES [dbo].[Fridge] ([FridgeID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BucketFridge] CHECK CONSTRAINT [FK_BucketFridge_Fridge]
GO
```

/****** Object: ForeignKey [FK_FridgeTruck_Fridge] Script Date: 05/21/2013 23:33:01
*****/

```
ALTER TABLE [dbo].[FridgeTruck] WITH CHECK ADD CONSTRAINT
[FK_FridgeTruck_Fridge] FOREIGN KEY([FridgeID])
REFERENCES [dbo].[Fridge] ([FridgeID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[FridgeTruck] CHECK CONSTRAINT [FK_FridgeTruck_Fridge]
GO
```

/****** Object: ForeignKey [FK_FridgeTruck_Truck] Script Date: 05/21/2013 23:33:01
*****/

```
ALTER TABLE [dbo].[FridgeTruck] WITH CHECK ADD CONSTRAINT
[FK_FridgeTruck_Truck] FOREIGN KEY([TruckID])
REFERENCES [dbo].[Truck] ([TruckID])
ON UPDATE CASCADE
```

```
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[FridgeTruck] CHECK CONSTRAINT [FK_FridgeTruck_Truck]
GO
/***** Object: ForeignKey [FK_Route_Driver]  Script Date: 05/21/2013 23:33:01 *****/
ALTER TABLE [dbo].[Route] WITH CHECK ADD CONSTRAINT [FK_Route_Driver]
FOREIGN KEY([DriverID])
REFERENCES [dbo].[Driver] ([DriverID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Route] CHECK CONSTRAINT [FK_Route_Driver]
GO
/***** Object: ForeignKey [FK_Route_Truck]  Script Date: 05/21/2013 23:33:01 *****/
ALTER TABLE [dbo].[Route] WITH CHECK ADD CONSTRAINT [FK_Route_Truck]
FOREIGN KEY([TruckID])
REFERENCES [dbo].[Truck] ([TruckID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Route] CHECK CONSTRAINT [FK_Route_Truck]
GO
```