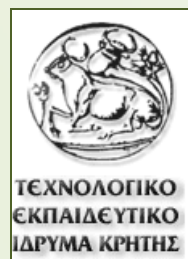


Τμήμα Εφαρμοσμένης Πληροφορικής και πολυμέσων

Πτυχιακή Εργασία

Δημιουργία οδηγού (wizard) για τη χωρική περιγραφή μιας εικόνας και την τεκμηρίωση του περιεχομένου σε xml.

Επιβλέπων Καθηγητής: Μαλάμος Αθανάσιος



Οκέκε Σταυρούλα 1328

Ηράκλειο 2008 - 2009

Περιεχόμενα

Ευχαριστίες.....	4
Κεφάλαιο 1.....	5
Εισαγωγή. Περιγραφή χώρου και αποτύπωση της δομημένης πληροφορίας σε αρχείο Xml.....	5
1.1 Σκοπός πτυχιακής εργασίας.....	7
Κεφάλαιο 2. Λειτουργικότητα.....	9
Κεφάλαιο 3.....	16
3.1 Επεξήγηση της γραφικής δομής του παραθύρου του οδηγού.....	16
3.2 Επεξήγηση Περιήγησης.....	16
3.3 Επεξήγηση μεθόδων setBackButtonEnabled() και setNextButtonEnabled().....	21
3.4 Comboboxes που φορτώνουν από το Configuration xml.....	22
3.5 Επεξήγηση ActionListener του κουμπιού browse και των κλάσεων ImageFilter , ImagePreview και Utils...34	
3.5.1 ActionListener του κουμπιού browse.....	34
3.5.2 Η κλάση ImageFilter.....	35
3.5.3 Επεξήγηση της κλάσης ImagePreview.....	35
3.5.4 Επεξήγηση της κλάσης Utils.....	37
3.6 Επεξήγηση των ActionListeners των κουμπιών επιλογής χρώματος (jColorChoosers).....	38
3.7 Επεξήγηση της κλάσης RoomObjects.....	38
3.8 Επεξήγηση του ActionListener των κουμπιών Add των αντικειμένων του δωματίου.....	39
3.8.1 ActionListener κουμπιού Add για τα αντικείμενα του Vector furniture.....	39
3.8.2 ActionListener κουμπιού Add για τα αντικείμενα του Vector accessories.....	41
3.9 Επεξήγηση μεθόδων doHex, checkMatId , checkStyleId, checkShapId και checkFloorId.....	42
3.9.1 Η doHex.....	42
3.9.2 checkMatId και checkStyleId.....	42
3.10 Επεξήγηση του ActionListener του κουμπιού finish.....	46
3.10.1 Επεξήγηση της μεθόδου doUpdate.....	50
3.10.2 Επεξήγηση της μεθόδου doUpdateVectors.....	54
Κεφάλαιο 4. Παράδειγμα χρήσης της εφαρμογής.....	60
4.1 Το αρχείο Template (XML).....	60
4.2 Το αρχείο που προκύπτει μετά το τρέξιμο της εφαρμογής.....	66
Κεφάλαιο 5. Παράρτημα.....	73
5.1 Xml.....	73
5.2 XPath.....	74
5.3 XQuery.....	74
5.4 Exist.....	75
Κεφάλαιο 6. Βιβλιογραφία.....	76
6.1 Πηγές.....	76

Ευχαριστίες

Ευχαριστώ τον κ. Μαλάμο για την πολύτιμη βοήθεια του και τις οδηγίες του, τους Γιώργο Μαμάκη και Εύη Σίμπα που καθ' όλη την διάρκεια της πτυχιακής εργασίας με βοήθησαν και ήταν σε επικοινωνία μαζί μου, όπως επίσης και όλους τους συμφοιτητές μου στο εργαστήριο MClab που με βοήθησαν και συνέβαλαν στην ολοκλήρωση της εργασίας αυτής.

Κεφάλαιο 1.

Εισαγωγή.

Περιγραφή χώρου και αποτύπωση της δομημένης πληροφορίας σε αρχείο Xml

Στόχος της εργασίας αυτής είναι η επίτευξη μιας πλήρους περιγραφής ενός χωρού σε μορφή δομημένου αρχείου, συγκεκριμένα αρχείου Xml. Το αρχείο αυτό που προκύπτει μπορεί να βρεί χρήση σε πολλές εφαρμογές όπου θα χρησιμοποιηθεί για την μετέπειτα ψηφιακή αναπαράσταση του χωρού, και θα αποτελεί τον «σκελετό» του, περιέχοντας όλες τις πληροφορίες για αυτόν και για τα αντικείμενα που βρίσκονται μέσα σε αυτόν.

Οι εφαρμογές που θα χρησιμοποιήσουν τα αρχεία αυτά, αρχιτεκτονικού ή άλλου σχεδιαστικού ενδιαφέροντος αντλούν από το αρχείο όλες τις απαραίτητες πληροφορίες για το χώρο , από τα λεπτομερή στοιχεία της περιγραφής του, όπως για παράδειγμα το χρώμα και το υλικό των τοίχων, του πατώματος, μέχρι το χρώμα και το υλικό των αντικειμένων που πιθανόν να βρίσκονται πάνω σε αυτά. Έτσι γίνεται δυνατή η αναδόμηση και αναπαράσταση του χωρού από το εκάστοτε πρόγραμμα ή εφαρμογή που θα το χρησιμοποιήσει.

Τα Xml αρχεία είναι ιδανικά για την αποτύπωση της περιγραφής αυτής. (Αναλυτικότερα για την Xml γλώσσα, στο παράρτημα στην παράγραφο 5.1)

Η χρήση του οδηγού που δημιουργήθηκε, ουσιαστικά αποτελεί το μέσον για να μετατραπεί η οπτική πληροφορία που αποτελεί ένα δωμάτιο, σε ψηφιακή πληροφορία (αρχείο) που μπορεί να «διαβάσει» μια εφαρμογή. Και στη συνέχεια να την χρησιμοποιήσει.

Το αρχείο σαν δομή αποτελεί όλοκληρο το δωμάτιο. Το πρώτο node του αρχείου, το header, περιλαμβάνει πληροφορίες που θα φανούν χρήσιμες στην εφαρμογή, όπως η τοποθεσία του αρχείου και τα στοιχεία του συγκεκριμένου project, όπως επίσης και πληροφορίες για τις μονάδες μέτρησης. Μετά ξεκινάει το tag “room”, το οποίο απαρτίζεται από τους κόμβους με όλα τα στοιχεία του δωματίου.

Για παράδειγμα ας εξετάσουμε το node του ταβανιού «ceiling».

```
<ceiling>
  <model3D>
    <coordinates> </coordinates>
    <path> </path>
  </model3D>
  <shape> </shape>
  <dimensions> </dimensions>
  <style styleIndexName=" " styleIndexURL=" "> </style>
  <material>
    <materialType materialIndexName=" " materialIndexURL=" "> </materialType>
    <pattern patternIndexName=" " patternIndexURL=" "> </pattern>
  </material>
  <ColorPalette ColorPaletteIndexName=" " ColorPaletteIndexURL=" "> </ColorPalette>
  <lightSource>
    <model3D>
      <coordinates> </coordinates>
```

```

    <path> </path>
  </model3D>
</dimensions> </dimensions>
<RelativePosition ReferenceObject=" "></RelativePosition>
<lightSourcePalletteIndex
  lightSourcePalletteIndexName=" " lightSourcePalletteIndexURL=" "> </lightSourcePalletteIndex>
</lightSource>
<Construction>
  <model3D>
    <coordinates </coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  </dimensions> </dimensions>
  <RelativePosition ReferenceObject=" "> </RelativePosition>
  <style styleIndexName=" " styleIndexURL=" "> </style>
  <material>
    <materialType
      materialIndexName=" " materialIndexURL=" "> </materialType>
    <pattern patternIndexName=" " patternIndexURL=" "> </pattern>
  </material>
  <ColorPalette ColorPaletteIndexName=" " ColorPaletteIndexURL=" "> </ColorPalette>
</Construction>
</ceiling>

```

Η δομή του ταβανιού περιγράφεται πλήρως, με tag για διαστάσεις υλικό, στυλ. Και περιλαμβάνει κόμβους – παιδιά για να ολοκληρωθεί επιμέρους. Η χρήση των attributes συμβάλει σημαντικά στη λεπτομέρεια και την αρτιότητα της περιγραφής. Στοιχεία όπως κωδικές ονομασίες και url στοιχείων ή λοιπών αρχείων (πχ x3D μοντέλα) περιλαμβάνονται στα αντίστοιχα tag, διαθέτοντας όλες τις πληροφορίες στην εφαρμογή του θα χρησιμοποιήσει το xml αρχείο αργότερα.

1.1 Σκοπός πτυχιακής εργασίας

Σκοπός της εργασίας αυτής είναι η υλοποίηση ενός οδηγού σε java, μέσω του οποίου περιγράφεται λεπτομερώς ένα δωμάτιο. Μέσω του οδηγού αυτού μπορούμε να περιγράψουμε γενικά τον χώρο (ως προς το στυλ και τη διάθεση) , αλλά και ειδικότερα, σε σημεία όπως τα χρώματα των τοίχων, τα αντικείμενα, τις πόρτες , τα παράθυρα και τις σκάλες.

Με την ολοκλήρωση του οδηγού, το δωμάτιο έχει περιγραφεί πλήρως και η περιγραφή αυτή συμπύσσεται σε ένα αρχείο xml το οποίο αποθηκεύεται σε μια βάση δεδομένων (exist).

Ο οδηγός ολοκληρώνεται σε τρία βήματα και αποτελείται συνολικά από τρεις καρτέλες. Ο χρήστης κατά την περιήγησή του στον οδηγό έχει τη δυνατότητα να ανατρέξει ανα πάσα στιγμή σε οποιαδήποτε καρτέλα για να αλλάξει ή να επιθεωρήσει τις επιλογές του.

Η πρώτη καρτέλα αφορά τις γενικές επιλογές και τις πληροφορίες για το ταβάνι και τον τοίχο. Ο χρήστης τα περιγράφει λεπτομερώς, ορίζοντας τα υλικά και χρώματα για τις πόρτες, τα παράθυρα, τον κύριο φωτισμό του δωματίου και το ταβάνι.

Η δεύτερη καρτέλα αφορά το πάτωμα και τα έπιπλα του δωματίου. Για το πάτωμα οι επιλογές γίνονται όπως και πριν από τα comboboxes, ενώ για τα έπιπλα η διαδικασία είναι λίγο διαφορετική. Ανάλογα με τον τύπο του επίπλου, ο χρήστης επιλέγει τα χαρακτηριστικά του (type, material, frame material, style, color, frame color) από τα αντίστοιχα comboboxes και στη συνέχεια επιλέγει "add". Τότε προστίθεται το νέο έπιπλο στην περιγραφή του δωματίου και εμφανίζεται στη λίστα με τα έπιπλα του wizard, για να μπορεί ο χρήστης διαρκώς να έχει μια εικόνα για το ποια έπιπλα έχει προσθέσει.

Η τρίτη και τελευταία καρτέλα αφορά το χαλί και τα αντικείμενα που βρίσκονται στους τοίχους του δωματίου. Η προσθήκη των αντικειμένων γίνεται όπως έγινε και η προσθήκη των επίπλων νωρίτερα. Τα αντικείμενα είναι και πάλι χωρισμένα σε κατηγορίες, και υπάρχουν διαφορετικές επιλογές περιγραφής των χαρακτηριστικών τους, ανάλογα με την κατηγορία στην οποία ανήκουν.

Οι περιγραφικές επιλογές του οδηγού, τα στοιχεία δηλαδή των comboboxes κάθε φορά, ορίζονται από ένα εξωτερικό αρχείο xml το οποίο περιλαμβάνει όλα τα περιεχόμενά τους, και το οποίο μπορεί να αλλάζει κάθε φορά ανάλογα με τις ανάγκες της περιγραφής.

Συνοπτικά, ο τρόπος με τον οποίο θα επιτευχθεί η αποθήκευση του τελικού αρχείου στην exist ακολουθεί την εξής σειρά γεγονότων:

1. Δημιουργείται στον σκληρό δίσκο ένα αντίγραφο του αρχείου xml το οποίο λειτουργεί ως πρότυπο του τελικού. Το πρότυπο αρχείο βρίσκεται ήδη στην exist.
2. Δημιουργείται εκ νέου στην exist σε collection με όνομα final, το αντίγραφο.

3. Ανάλογα με τις επιλογές του χρήστη στον οδηγό θα πραγματοποιηθούν τα updates στο xml αντίγραφο δίνοντας μας έτσι το τελικό αρχείο.

Κεφάλαιο 2. Λειτουργικότητα

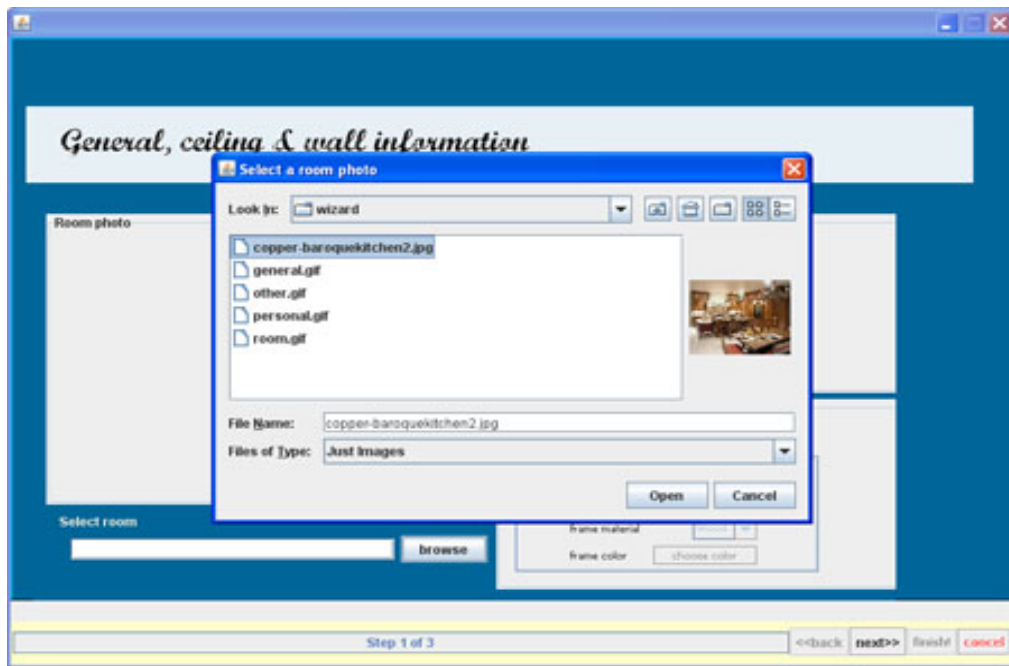
Ξεκινώντας την εφαρμογή ο χρήστης επιλέγει *browse* , για να βρει την φωτογραφία του δωματίου για το οποίο θα γίνει η περιγραφή.

The screenshot shows a software window titled "General, ceiling & wall information". The interface is divided into several sections:

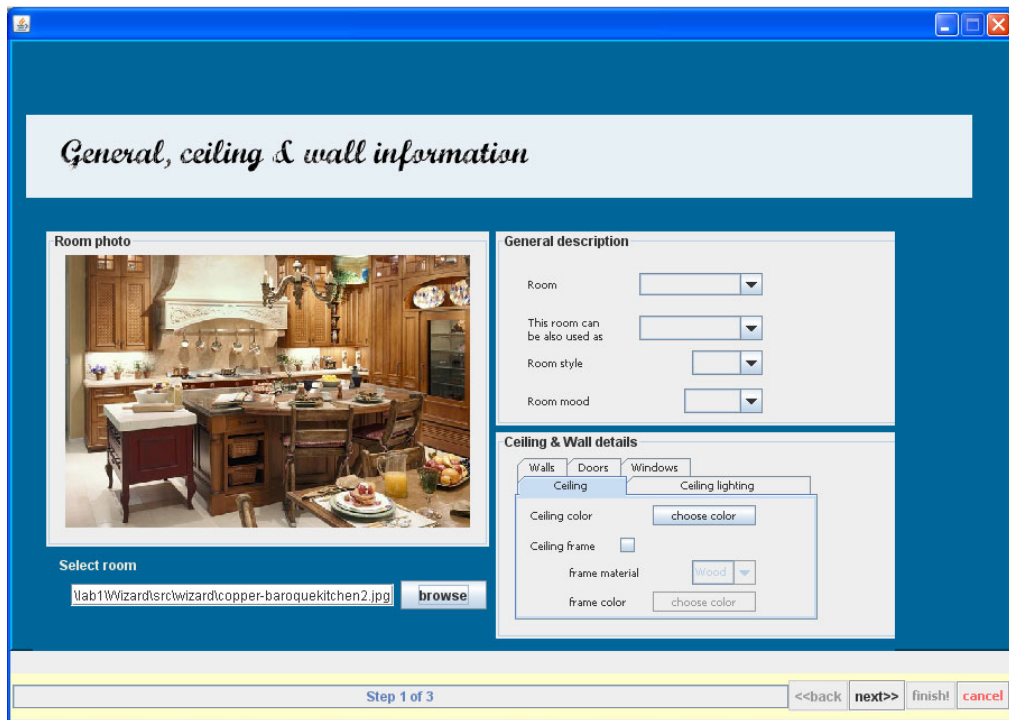
- Room photo:** A large empty rectangular area for displaying a room photo.
- General description:** A section containing four dropdown menus:
 - Room
 - This room can be also used as
 - Room style
 - Room mood
- Ceiling & Wall details:** A section with tabs for "Walls", "Doors", and "Windows". The "Ceiling" tab is active, showing:
 - Ceiling lighting:** A sub-section with a "choose color" button.
 - Ceiling color:** A "choose color" button.
 - Ceiling frame:** A checkbox.
 - frame material:** A dropdown menu currently set to "Wood".
 - frame color:** A "choose color" button.
- Select room:** A text input field with a "browse" button next to it.

At the bottom of the window, there is a progress bar indicating "Step 1 of 3" and navigation buttons: "<<back", "next>>", "finish!", and "cancel".

Στην συνέχεια στο παράθυρο που εμφανίζεται επιλέγει την εικόνα του δωματίου που τον ενδιαφέρει να περιγράψει.

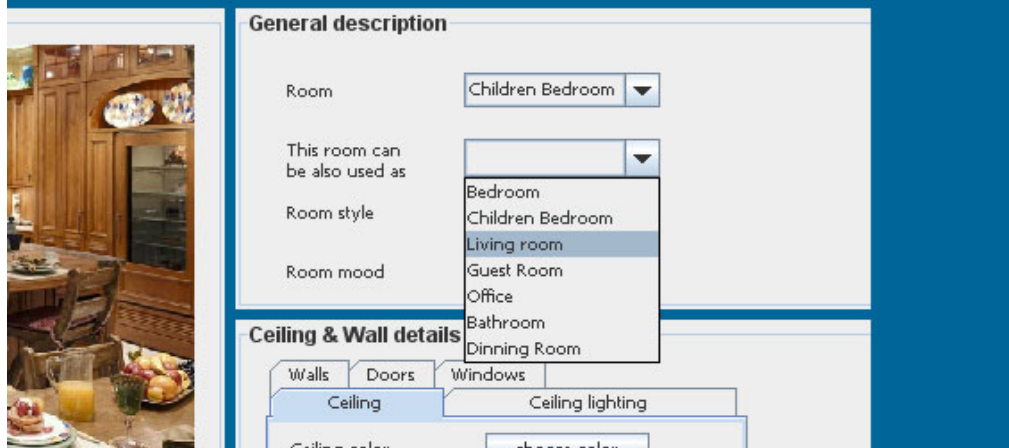


Και επιλέγει Open για να ανοίξει η εικόνα.



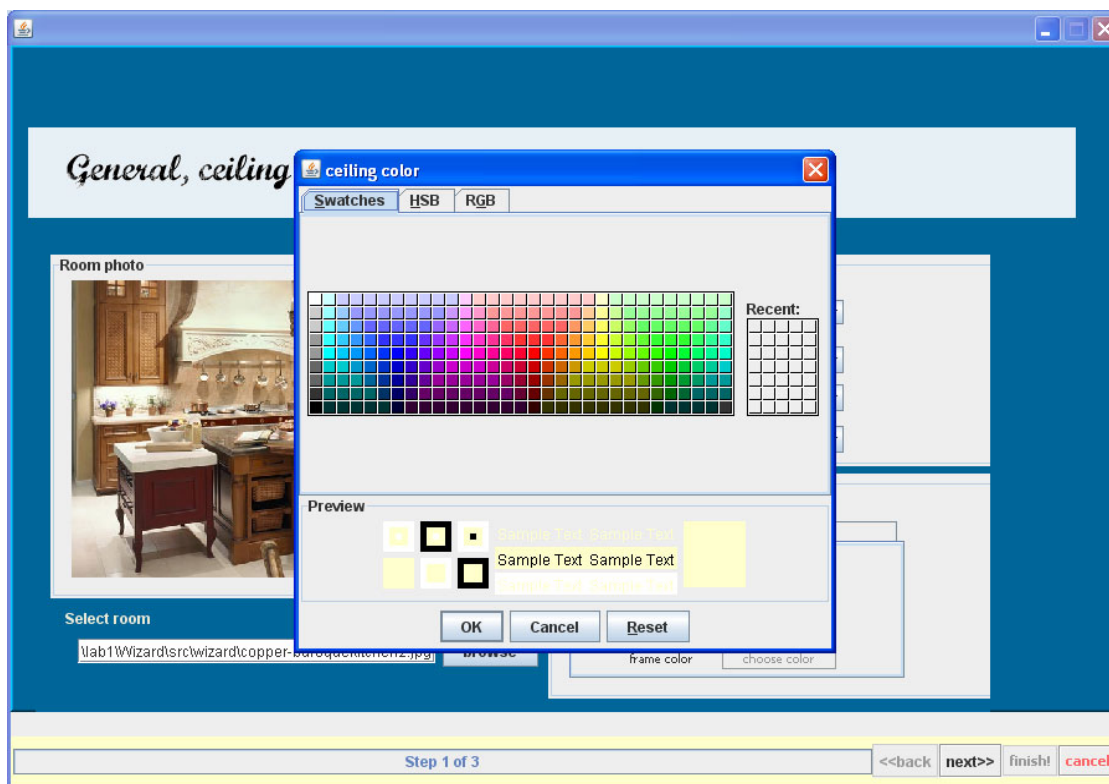
Στο πάνω δεξιά πάνελ του οδηγού βρίσκονται τέσσερα Comboboxes γενικής επιλογής όσον αφορά το είδος, το στυλ και την διάθεση του δωματίου. Ο Χρήστης μπορεί να επιλέξει με το ποντίκι.

el information

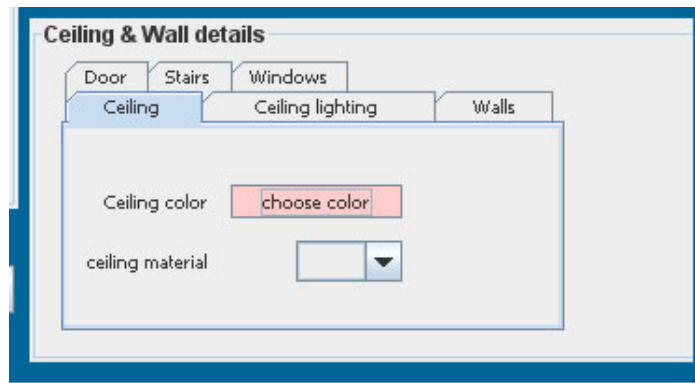


Το μέρος που αφορά τις επιλογές των τοίχων και του ταβανιού βρίσκεται ακριβώς από κάτω, και απαρτίζεται από πέντε καρτέλες. Μία για τοίχους, μια για το ταβάνι, για τις πόρτες και τέλος μία για τα παράθυρα.

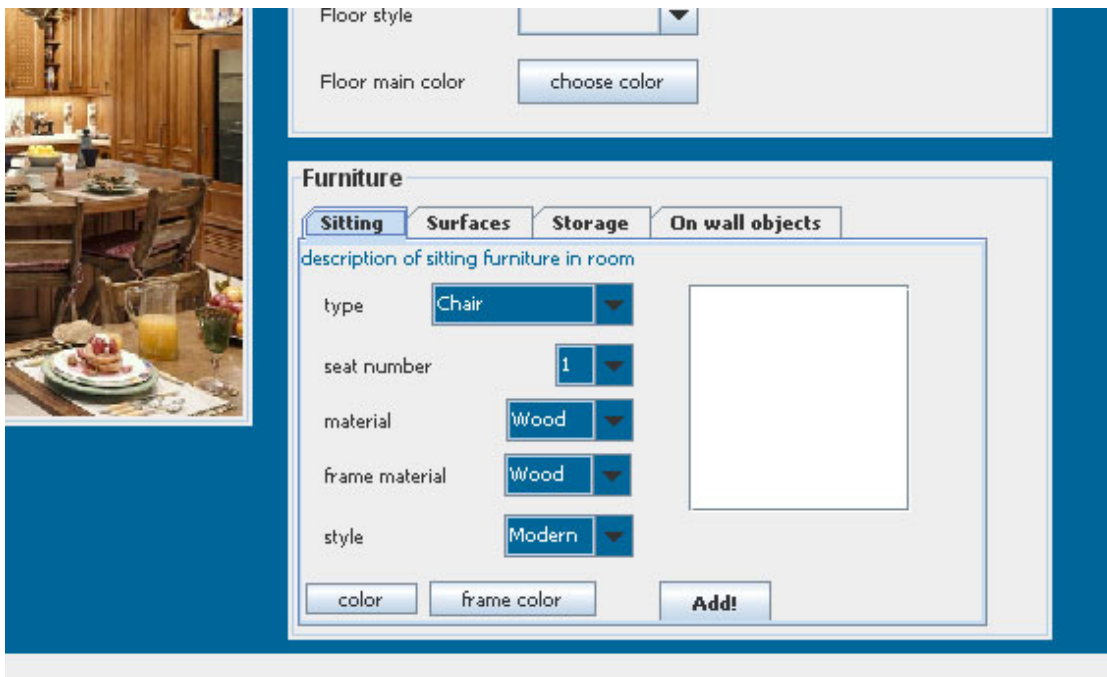
Πατώντας πάνω στο κουμπί επιλογής χρώματος για το ταβάνι, όπως συμβαίνει και με όλα τα κουμπιά επιλογής χρώματος στον wizard, εμφανίζεται ένα παράθυρο επιλογής χρώματος.



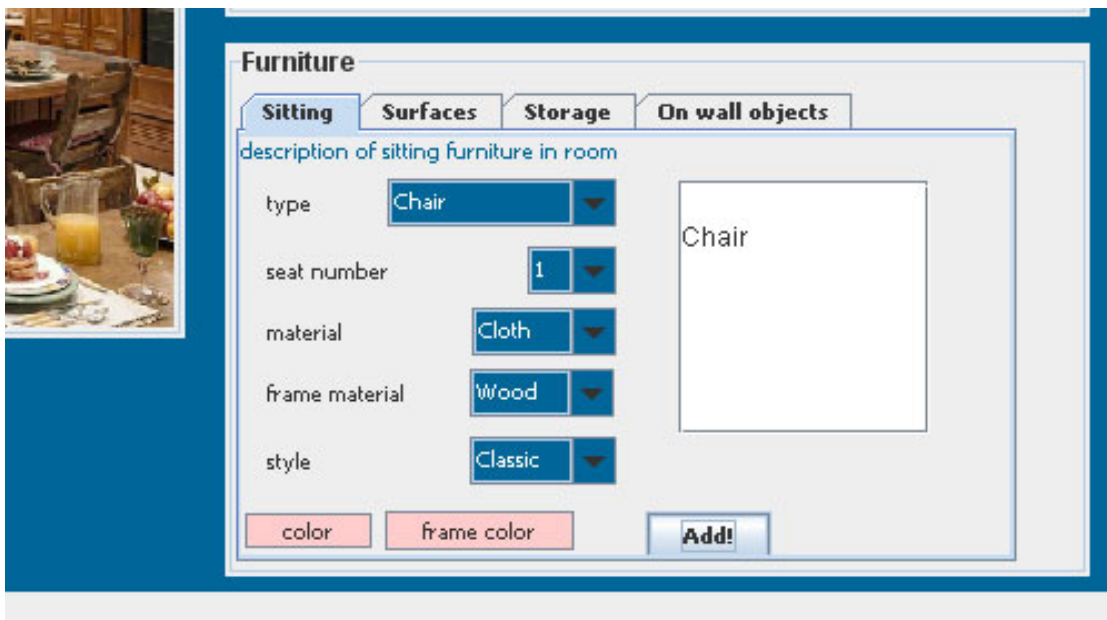
Ο χρήστης μπορεί να επιλέξει το χρώμα ή να το βρει χρησιμοποιώντας τιμές RGB. Με το πάτημα του κουμπιού OK το κουμπί παίρνει το επιλεγμένο χρώμα.



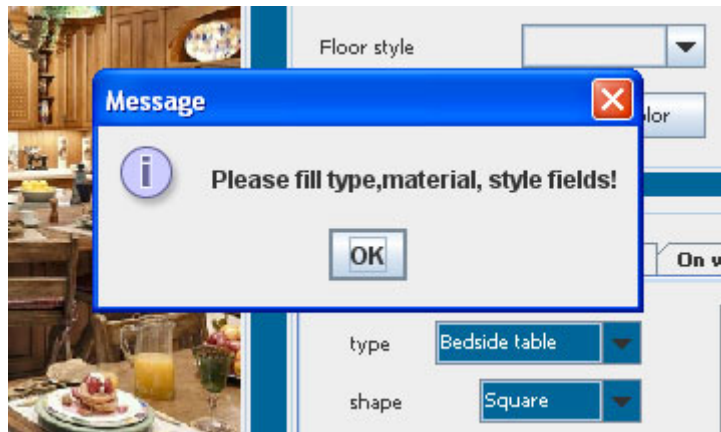
Στις καρτέλες περιγραφής των αντικειμένων υπάρχουν επιλογές ανά καρτέλα κατηγορίας αντικειμένων.



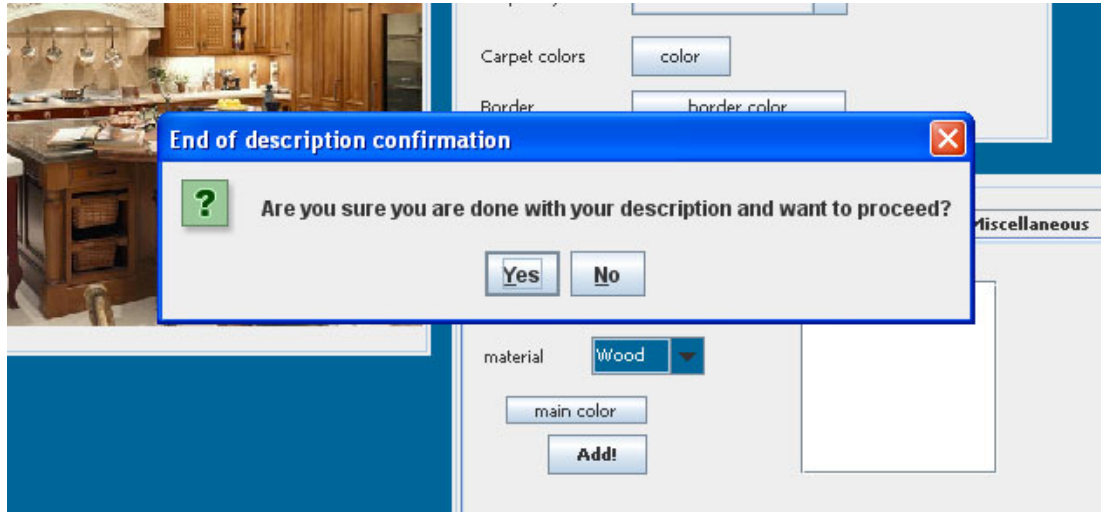
Ο χρήστης επιλέγει τα χαρακτηριστικά του αντικειμένου και επιλέγει Add. Το αντικείμενο προστίθεται στη λίστα.



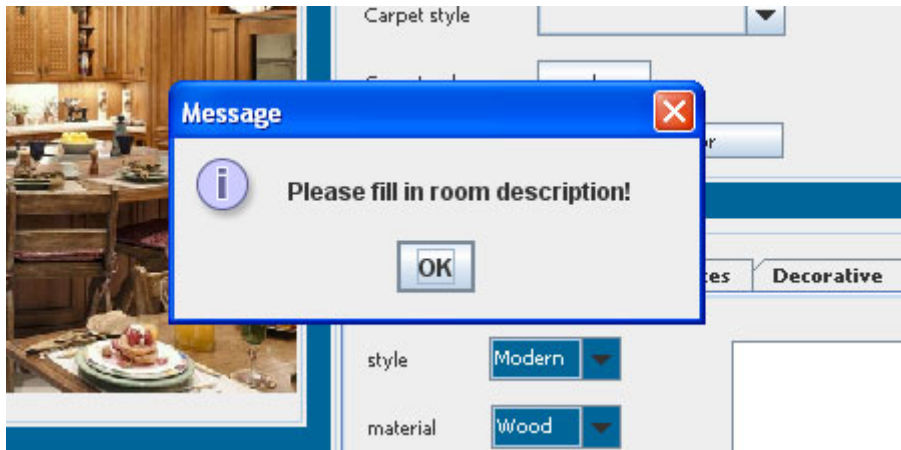
Σε περίπτωση που δεν συμπληρωθεί κάποιο πεδίο. Εμφανίζεται ένα μήνυμα. Αυτό ισχύει για όλες τις καρτέλες αντικειμένων του οδηγού.



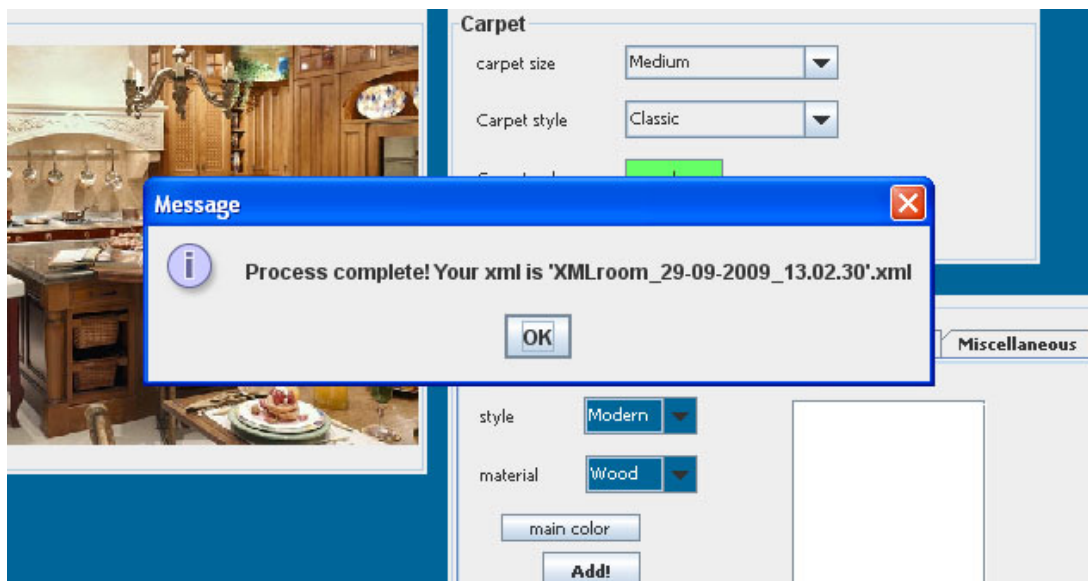
Με το πάτημα του κουμπιού finish για ολοκλήρωση της διαδικασίας, ζητείται από την εφαρμογή επιβεβαίωση για να προχωρήσει.



Μόλις ο χρήστης επιβεβαιώσει, εμφανίζεται μήνυμα ειδοποίησης σε περίπτωση που κάποια πεδία είναι άδεια. Για παράδειγμα εάν ξεχάσει κάποιο από τα πεδία γενικής περιγραφής του δωματίου.



Και μόλις η διαδικασία ολοκληρωθεί επιτυχώς. Εμφανίζεται μήνυμα.

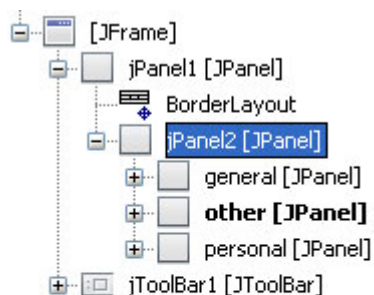


Το νέο αρχείο βρίσκεται στην exist στο collection final.

Κεφάλαιο 3

3.1 Επεξήγηση της γραφικής δομής του παραθύρου του οδηγού.

Ο οδηγός αποτελείται από την κλάση Wizard1. Το JFrame περιλαμβάνει ένα πάνελ (BorderLayout) πάνω στο οποίο έχουν τοποθετηθεί, άλλο ένα πάνελ (περιέχει τις καρτέλες επιλογών του οδηγού) και το JToolBar που φιλοξενεί τα κουμπιά περιήγησης. Οι καρτέλες (JPanels) του οδηγού ονομάζονται General, Other και Personal και είναι τοποθετημένες όπως φαίνεται παρακάτω.



Ο οδηγός πρόκειται για ένα αντικείμενο τύπου Wizard1 το οποίο έχει διαστάσεις 900 x 640 px και τοποθετείται στο κέντρο της οθόνης κάθε φορά.

```
Wizard1 c=new Wizard1();
int frameWidth = 900;
int frameHeight = 640;
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
c.setBounds((int) (screenSize.getWidth() - frameWidth)/2,
(int)(screenSize.getHeight()-frameHeight)/2, frameWidth, frameHeight);
c.setVisible(true);
```

3.2 Επεξήγηση Περιήγησης.

Ο τρόπος διάταξης των JPanel είναι CardLayout και επιτρέπει την εναλλαγή των εμφανιζόμενων κάθε πάνελ με ένα σετ εντολών.

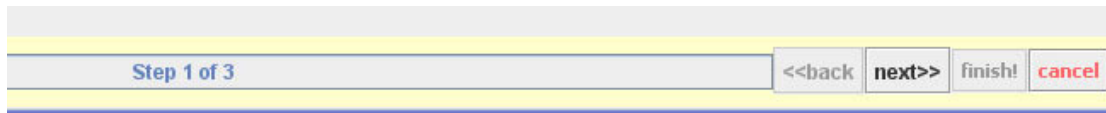
Ο κώδικας δημιουργίας του CardLayout:

```
Η δήλωση:
private CardLayout card,floorcard;
```


Δημιουργία:

```
card = new CardLayout();  
jPanel2.setLayout(card);  
floorcard= new CardLayout();
```

Στη βάση του παραθύρου βρίσκεται το JToolBar πάνω στο οποίο βρίσκονται τα κουμπιά περιήγησης και μία JProgressBar.



Στην ProgressBar φαίνεται η πορεία ολοκλήρωσης του wizard.

Κώδικας μπάρας:

```
jProgressBar1.setMaximum(4); int valuebar=jProgressBar1.getValue();  
jProgressBar1.setIndeterminate(false);  
jProgressBar1.setValue(0);  
jProgressBar1.setStringPainted(true);  
jProgressBar1.setString("Step 1 of 3");
```

Οι ActionListener των κουμπιών. (για λόγους καλαισθησίας του CardLayout τα κουμπιά και comboboxes των μη ορατών πάνελ πρέπει να απενεργοποιούνται)
Ο κώδικας του κουμπιού back:

// Καθε φορά που πατάμε το κουμπί next της εφαρμογής η μεταβλητή next αυξάνεται κατά 1 και ανάλογα την τιμή που έχει πηγαίνει στην επομένη καρτέλα και γίνονται ενεργά και ανενεργά αντίστοιχα κάποια jButtonons και jLabels.

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    sum--;  
    if (sum==1)  
    {  
        card.first(jPanel2);  
        setBackButtonEnabled(false);  
        setNextButtonEnabled(true);  
        jButton2.setEnabled(false);  
        jTextField1.setEnabled(true);  
        jProgressBar1.setValue(valuebar-2);jProgressBar1.getValue();  
        valuebar=valuebar-2;  
        jProgressBar1.setString("Step 1 of 3");  
    }  
}
```

```

else if (sum==2)
{
card.previous(jPanel2);
jTextField1.setEnabled(false);
jComboBox11.setEnabled(false);
jButton6.setEnabled(true);
jComboBox16.setEnabled(true);
jComboBox15.setEnabled(true);
jComboBox5.setEnabled(true);
jComboBox6.setEnabled(true);
jComboBox7.setEnabled(true);
jComboBox8.setEnabled(true);
jComboBox13.setEnabled(true);
jButton3.setEnabled(true);
jButton7.setEnabled(true);
jButton8.setEnabled(true);
jButton10.setEnabled(true);
jLabel26.setVisible(true);
jLabel41.setVisible(false);
setBackButtonEnabled(true);
setNextButtonEnabled(true);
jButton2.setEnabled(false);
jProgressBar1.setValue(valuebar-2);
jProgressBar1.getValue();
valuebar=valuebar-2;
jProgressBar1.setString("Step 2 of 3");

}

else if(sum==3)
{
card.last(jPanel2);
jComboBox27.setEnabled(true);
jComboBox28.setEnabled(true);
jComboBox11.setEnabled(true);
jComboBox29.setEnabled(true);
jPanel40.setVisible(true);
jComboBox5.setEnabled(true);
jComboBox6.setEnabled(true);
jButton21.setEnabled(true);
jLabel26.setVisible(false);
jLabel41.setVisible(true);
setBackButtonEnabled(true);
setNextButtonEnabled(false);

jButton2.setEnabled(true);
jProgressBar1.setValue(valuebar+2);

```

```

        progressBar1.getValue();
        progressBar1.setString("Step 3 of 3");

    }
    else if(sum<1)
    {
        sum=1;
    }
}

```

Ο κώδικας του κουμπιού Next:

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    sum++;
    if (sum==1)
    {
        card.first(jPanel2);
        setBackButtonEnabled(false);
        setNextButtonEnabled(true);
        jButton2.setEnabled(false);
        progressBar1.getValue();
        progressBar1.setString("Step 1 of 3");
        jTextField1.setEnabled(true);
    }

    else if (sum==2)
    {
        card.next(jPanel2);
        jTextField1.setEnabled(false);
        jButton6.setEnabled(true);
        jComboBox16.setEnabled(true);
        jComboBox15.setEnabled(true);
        jComboBox11.setEnabled(false);
        jComboBox5.setEnabled(true);
        jComboBox6.setEnabled(true);
        jComboBox7.setEnabled(true);
        jComboBox8.setEnabled(true);
        jComboBox13.setEnabled(true);

        jLabel26.setVisible(true);
        jLabel41.setVisible(false);

        jButton3.setEnabled(true);
        jButton7.setEnabled(true);
        jButton8.setEnabled(true);
        jButton10.setEnabled(true);
    }
}

```

```

setBackButtonEnabled(true);
setNextButtonEnabled(true);
jButton2.setEnabled(false);
jProgressBar1.setValue(valuebar+2);
jProgressBar1.getValue();
valuebar=valuebar+2;
jProgressBar1.setString("Step 2 of 3");
}

else if(sum==3)
{
    card.last(jPanel2);
jComboBox11.setEnabled(true);
jComboBox29.setEnabled(true);
jComboBox5.setEnabled(true);
jComboBox6.setEnabled(true);
jComboBox27.setEnabled(true);
jComboBox28.setEnabled(true);
jButton24.setEnabled(true);
jButton25.setEnabled(true);
jButton21.setEnabled(true);
jButton23.setEnabled(true);
jButton22.setEnabled(true);
jButton21.setEnabled(true);
jButton28.setEnabled(true);

jLabel26.setVisible(false);
jLabel41.setVisible(true);

    setBackButtonEnabled(true);
    setNextButtonEnabled(false);
    jButton2.setEnabled(true);
    jProgressBar1.setValue(valuebar+2);jProgressBar1.getValue();
    valuebar=valuebar+2;
    jProgressBar1.setString("Step 3 of 3");
}
else if(sum>3)
{
    sum=3;
}
}

```

Το κουμπί cancel :

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    int n= JOptionPane.showConfirmDialog(null,"Are you sure you want to cancel?",  
    "Cancel and Exit the wizard", JOptionPane.YES_NO_OPTION);  
    if (n== JOptionPane.YES_OPTION) {System.exit(0);}  
}
```

Για τις ανάγκες ενεργοποίησης και απενεργοποίησης των κουμπιών περιήγησης καλούνται οι μέθοδοι `setBackButtonEnabled()` και `setNextButtonEnabled()` .

3.3 Επεξήγηση μεθόδων `setBackButtonEnabled()` και `setNextButtonEnabled()`

```
void setBackButtonEnabled(boolean b) {  
    jButton4.setEnabled(b);  
}  
  
void setNextButtonEnabled(boolean b) {  
    jButton1.setEnabled(b);  
}
```

3.4 Comboboxes που φορτώνουν από το Configuration xml

Όλα τα jComboBox της εφαρμογής «τραβάνε» τις τιμές τους από ένα αρχείο xml.

Το xml βρίσκεται στη θέση C:/Configuration , και έχει την παρακάτω μορφή:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<Combo_Box>
```

```
<Drop_Down name="majorfunctionality">
```

```
<Item id="1">Bedroom</Item>
```

```
<Item id="2">Children Bedroom</Item>
```

```
<Item id="3">Living room</Item>
```

```
<Item id="4">Guest Room</Item>
```

```
<Item id="5">Office</Item>
```

```
<Item id="6">Bathroom</Item>
```

```
<Item id="7">Dinning Room</Item>
```

```
</Drop_Down>
```

```
<Drop_Down name="minorfunctionality">
```

```
<Item id="1">Bedroom</Item>
```

```
<Item id="2">Children Bedroom</Item>
```

```
<Item id="3">Living room</Item>
```

```
<Item id="4">Guest Room</Item>
```

```
<Item id="5">Office</Item>
```

```
<Item id="6">Bathroom</Item>
```

```
<Item id="7">Dinning Room</Item>
```

```
</Drop_Down>
```

```
<Drop_Down name="style">
```

```
<Item id="1">Modern</Item>
```

```
<Item id="2">Classic</Item>
```

```
<Item id="3">Retro</Item>
```

```
<Item id="4">Disco</Item>
```

```
</Drop_Down>
```

```
<Drop_Down name="mood">
```

```
<Item id="1">Joy</Item>
```

```
<Item id="2">Calm</Item>
```

```
<Item id="3">Energetic</Item>
```

```
</Drop_Down>
```

```
<Drop_Down name="onwallobj">
```

```
<Item id="1">Mirror</Item>
```

```
<Item id="2">Frame</Item>
<Item id="3">Painting</Item>
<Item id="4">Hook</Item>
<Item id="5"></Item>
</Drop_Down>

<Drop_Down name="storagetype">
<Item id="1">Shelf</Item>
<Item id="2">Armoire</Item>
<Item id="3">Drawer</Item>
<Item id="4">Other</Item>
</Drop_Down>

<Drop_Down name="shape">
<Item id="1">Square</Item>
<Item id="2">Rectangular</Item>
<Item id="3">Circular</Item>
<Item id="4">Polygonal</Item>
<Item id="5">Other</Item>
</Drop_Down>

<Drop_Down name="material">
<Item id="1">Wood</Item>
<Item id="2">Marble</Item>
<Item id="3">Mosaic</Item>
<Item id="4">Tile</Item>
<Item id="5">none</Item>
</Drop_Down>

<Drop_Down name="Label9">
<Item id="1">Single</Item>
<Item id="2">Couple</Item>
</Drop_Down>

<Drop_Down name="size">
<Item id="1">Small</Item>
<Item id="2">Medium</Item>
<Item id="3">Large</Item>
</Drop_Down>

<Drop_Down name="sittingtype">
<Item id="1">Sofa</Item>
<Item id="2">Chair</Item>
<Item id="3">Recliner</Item>
<Item id="4">Bed</Item>
<Item id="5">Stool</Item>
</Drop_Down>
```

```
<Drop_Down name="surfacetype">
<Item id="1">Table</Item>
<Item id="2">Coffee Table</Item>
</Drop_Down>
```

```
<Drop_Down name="doortype">
<Item id="1">Internal</Item>
<Item id="2">External</Item>
</Drop_Down>
```

```
<Drop_Down name="miscobj">
<Item id="1">Fish Tank</Item>
<Item id="2">Coat Hook</Item>
<Item id="3">Sculpture</Item>
<Item id="4">Gnome statue</Item>
<Item id="5">Feng shui fountain</Item>
</Drop_Down>
```

```
<Drop_Down name="wallobj">
<Item id="1">Fish Tank</Item>
<Item id="2">Coat Hook</Item>
<Item id="3">Sculpture</Item>
<Item id="4">Gnome statue</Item>
<Item id="5">Feng shui fountain</Item>
</Drop_Down>
```

```
<Drop_Down name="decorativetype">
<Item id="1">Antique</Item>
<Item id="2">Painting</Item>
<Item id="3">Statue</Item>
<Item id="4">Lamp</Item>
<Item id="5">Vase</Item>
</Drop_Down>
```

```
<Drop_Down name="electricaltype">
<Item id="1">Computer</Item>
<Item id="2">TV</Item>
<Item id="3">Hi-fi</Item>
<Item id="4">Playstation</Item>
</Drop_Down>
```

```
<Drop_Down name="stairtype">
<Item id="1">Eliptical</Item>
<Item id="2">Straight</Item>
<Item id="3">Circular</Item>
```



```
<Item id="4">Other</Item>
</Drop_Down>

<Drop_Down name="windowtype">
<Item id="1">Single</Item>
<Item id="2">Double</Item>
<Item id="3">Circular</Item>
<Item id="4">Other</Item>
</Drop_Down>
</Combo_Box>
```

Οι τιμές και τα attribute των πεδίων μπορούν να αλλάξουν ανάλογα με τις επιθυμίες για την υλοποίηση της εφαρμογής. Το μοναδικό σημείο που πρέπει να παραμείνει ως έχει είναι το attribute «name».

Ο κώδικας με τον οποίο γίνεται το γέμισμα:

```
try{
    try{
        try{
            try{
                try{
```

//Διαβάζουμε το Xml από το δίσκο που έχει τα δεδομένα μας χρησιμοποιώντας έναν FileReader. Έπειτα αποθηκεύουμε το αποτέλεσμα από αυτά που διαβάσαμε σε μια μεταβλητή τύπου String.

```
File f = new File("C:\\Configuration\\Combobox.xml");
FileReader rd = new FileReader(f);
char[] buf = new char[(int)f.length()];
rd.read(buf);
patternStr = new String(buf);
```

//Περνάμε τα δεδομένα σε μια μεταβλητή τύπου InputStream και δημιουργούμε ένα αντικείμενο τύπου DocumentBuilderFactory. Μέσω αυτού δημιουργούμε ένα αντικείμενο τύπου docBuilder και μέσω αυτού κάνουμε parse τα δεδομένα μας σε ένα αντικείμενο τύπου Document και με όνομα myXML.

```
InputStream is = new ByteArrayInputStream(patternStr.getBytes("utf-8"));
DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder docBuilder =docBuilderFactory.newDocumentBuilder();
Document myXML =docBuilder.parse(is);
myXML.normalize();
```

//Μεσω του myXML παίρνουμε το DocumentElement του XML μας στην περίπτωση μας το <Combo_Box> και το αποθηκεύουμε στην μεταβλητή myXMLElement.Έπειτα μέσω αυτού παίρνουμε μια λίστα με τα Elements που έχουν tagName Drop_Down και τα αποθηκεύουμε σε μια μεταβλητή τύπου NodeList.

```
Element myXMLElement = myXML.getDocumentElement();  
  
NodeList commonList = myXMLElement.getElementsByTagName("Drop_Down");  
  
int i=0;
```

//Με βάση το μέγεθος της NodeList φτάνουμε στο attribute του κάθε tag με όνομα Drop_Down και παίρνουμε την τιμή του.

```
while (i<commonList.getLength())  
{  
    Vector<String> ww=new Vector<String>();  
    Node commonNode = commonList.item(i);  
  
    Element commonElement = (Element)commonNode;  
    String b=(String)commonElement.getAttribute("name");
```

//Ανάλογα το όνομα που θα έχει κάθε φορά η μεταβλητή b μπαίνει στο κατάλληλο loop και παίρνουμε μια λίστα με τα Elements που έχουν tagName Item και τα αποθηκεύουμε σε μια μεταβλητή τύπου NodeList.

```
    if (b.equalsIgnoreCase("majorfunctionality")){  
        NodeList nodeList1=commonElement.getElementsByTagName("Item");  
        int k=0;  
        while (k<nodeList1.getLength()){  
            Node node1 = nodeList1.item(k);  
            Element element1 = (Element)node1;  
            Node myNode = node1.getNextSibling();  
            myNode=element1.getFirstChild();  
            String a =myNode.getNodeValue();  
            Element elem = (Element)nodeList1.item(k);  
            String id =elem.getAttribute("id");  
            jComboBox1.insertItemAt(a,k);  
            k++;  
        }  
    }  
  
    if (b.equalsIgnoreCase("minorfunctionality")){  
        NodeList nodeList1=commonElement.getElementsByTagName("Item");  
        int k=0;
```

```

while (k<nodeList1.getLength()){
    Node node1 = nodeList1.item(k);

    Element element1 = (Element)node1;
    Node myNode = node1.getNextSibling();
    myNode=element1.getFirstChild();
    String a =myNode.getNodeValue();
    Element elem = (Element)nodeList1.item(k);
    String id =elem.getAttribute("id");
    jComboBox2.insertItemAt(a,k);

    k++;

    }
}
if (b.equalsIgnoreCase("material")){
    NodeList nodeList1=commonElement.getElementsByTagName("Item");
    int k=0;
    while (k<nodeList1.getLength()){
        Node node1 = nodeList1.item(k);
        Element element1 = (Element)node1;
        Node myNode = node1.getNextSibling();
        myNode=element1.getFirstChild();
        String a =myNode.getNodeValue();

        jComboBox11.insertItemAt(a,k);
        jComboBox23.insertItemAt(a,k);
        jComboBox24.insertItemAt(a,k);
        jComboBox18.insertItemAt(a,k);
        jComboBox19.insertItemAt(a,k);
        jComboBox7.insertItemAt(a,k);
        jComboBox8.insertItemAt(a,k);
        jComboBox9.insertItemAt(a,k);
        jComboBox18.insertItemAt(a,k);
        jComboBox36.insertItemAt(a,k);
        jComboBox38.insertItemAt(a,k);
        jComboBox34.insertItemAt(a,k);
        jComboBox10.insertItemAt(a,k);
        jComboBox15.insertItemAt(a,k);
        k++;

    }
}

if (b.equalsIgnoreCase("style")){
    NodeList nodeList1=commonElement.getElementsByTagName("Item");
    int k=0;

```

```

while (k<nodeList1.getLength()){
    Node node1 = nodeList1.item(k);

    Element element1 = (Element)node1;
    Node myNode = node1.getNextSibling();
    myNode=element1.getFirstChild();
    String a =myNode.getNodeValue();
    jComboBox3.insertItemAt(a,k);
    jComboBox31.insertItemAt(a,k);
    jComboBox37.insertItemAt(a,k);
    jComboBox16.insertItemAt(a,k);
    jComboBox13.insertItemAt(a,k);
    jComboBox20.insertItemAt(a,k);
    jComboBox25.insertItemAt(a,k);
    jComboBox28.insertItemAt(a,k);
    jComboBox29.insertItemAt(a,k);
    k++;

    }
}

if (b.equalsIgnoreCase("mood")){
    NodeList nodeList1=commonElement.getElementsByTagName("Item");
    int k=0;
    while (k<nodeList1.getLength()){
        Node node1 = nodeList1.item(k);
        Element element1 = (Element)node1;
        Node myNode = node1.getNextSibling();
        myNode=element1.getFirstChild();
        String a =myNode.getNodeValue();
        jComboBox4.insertItemAt(a,k);
        k++;

        }
    }

if (b.equalsIgnoreCase("size")){
    NodeList nodeList1=commonElement.getElementsByTagName("Item");
    int k=0;
    while (k<nodeList1.getLength()){
        Node node1 = nodeList1.item(k);
        Element element1 = (Element)node1;
        Node myNode = node1.getNextSibling();
        myNode=element1.getFirstChild();
        String a =myNode.getNodeValue();
        jComboBox27.insertItemAt(a,k);
        k++;
    }
}

```

```

        }
    }
    if (b.equalsIgnoreCase("sittingtype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox5.insertItemAt(a,k);
            k++;
        }
    }
    if (b.equalsIgnoreCase("surfacetypetype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);

            Element element1 = (Element)node1;

            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox14.insertItemAt(a,k);
            k++;
        }
    }
    if (b.equalsIgnoreCase("shape")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox17.insertItemAt(a,k);
            jComboBox22.insertItemAt(a,k);
            k++;
        }
    }

```

```

        }
    }
    if (b.equalsIgnoreCase("storagetype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox21.insertItemAt(a,k);
            k++;
        }
    }
    if (b.equalsIgnoreCase("wallobj")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox26.insertItemAt(a,k);
            k++;
        }
    }
    if (b.equalsIgnoreCase("miscobj")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox40.insertItemAt(a,k);
            k++;
        }
    }
}
}

```

```

    }
    if (b.equalsIgnoreCase("decorativetype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox35.insertItemAt(a,k);
                k++;
            }
        }
    if (b.equalsIgnoreCase("electricaltype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox30.insertItemAt(a,k);
                k++;
            }
        }
    if (b.equalsIgnoreCase("doortype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox32.insertItemAt(a,k);
                k++;
            }
        }
    if (b.equalsIgnoreCase("stairtype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;

```

```

        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);

            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox39.insertItemAt(a,k);
            k++;
        }
    }
    if (b.equalsIgnoreCase("windowtype")){
        NodeList
nodeList1=commonElement.getElementsByTagName("Item");
        int k=0;
        while (k<nodeList1.getLength()){
            Node node1 = nodeList1.item(k);
            Element element1 = (Element)node1;
            Node myNode = node1.getNextSibling();
            myNode=element1.getFirstChild();
            String a =myNode.getNodeValue();
            jComboBox41.insertItemAt(a,k);
            k++;
        }
    }
    i++;
}

}
    catch (FileNotFoundException e){
    }
}
    catch (java.io.IOException e){
    }
}
    catch (ParserConfigurationException e){
    }
}
    catch (org.xml.sax.SAXException e){
    }
}
    catch(Exception e){
    e.printStackTrace();}

```


3.5 Επεξήγηση ActionListener του κουμπιού browse και των κλάσεων ImageFilter , ImagePreview και Utils.

3.5.1 ActionListener του κουμπιού browse.

Με το πάτημα του κουμπιού browse ανοίγει ένα παράθυρο αναζήτησης (JFileChooser) , για να βρει ο χρήστης την εικόνα που επιθυμεί , το οποίο δείχνει μικρή προεπισκόπηση στα δεξιά και δέχεται μόνο αρχεία εικόνων.

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    JFileChooser chooser=new JFileChooser();  
    chooser.setCurrentDirectory(new java.io.File("."));  
    chooser.setDialogTitle("Select a room photo");  
  
    chooser.setAcceptAllFileFilterUsed(false);  
    chooser.addChoosableFileFilter(new ImageFilter());  
    chooser.setAccessory(new ImagePreview(chooser));  
  
    int status = chooser.showOpenDialog(this);  
  
    if (status == JFileChooser.APPROVE_OPTION) {  
  
        File f = chooser.getSelectedFile();  
        String path = f.getPath();  
        jTextField1.setText(path);  
  
        // Το αρχείο που θα επιλέχθει δεν μπορεί να έχει οποιεσδήποτε διαστάσεις, γι'  
        // αυτό το λόγο γίνεται έλεγχος για το αν υπερβαίνει τα 500 pixel στο πλάτος, εάν ναι  
        // τότε του δίνεται η τιμή πλάτους 500.  
        ImageIcon tmpIcon = new ImageIcon(f.getPath());  
        if (tmpIcon != null) {  
  
            if (tmpIcon.getIconWidth() > 500) {  
                imageinstance = new ImageIcon(tmpIcon.getImage().  
                    getScaledInstance(500, -1,Image.SCALE_DEFAULT));  
            }  
            else { //no need to miniaturize  
                imageinstance = tmpIcon;  
            }  
        }  
    }  
  
    // Η επιλεγμένη εικόνα τοποθετείται πάνω στα j Label του wizard.  
    jLabel25.setIcon(imageinstance);  
    jLabel26.setIcon(imageinstance);  
    jLabel41.setIcon(imageinstance);  
}
```

```
}
```

3.5.2 Η κλάση ImageFilter

```
import java.io.File;
import javax.swing.filechooser.*;
// Τα επιτρεπτά είδη αρχείων που δέχεται το φίλτρο είναι gif, jpg, tiff, και png.
public class ImageFilter extends FileFilter {
    public boolean accept(File f) {
        if (f.isDirectory()) {
            return true;
        }

        String extension = Utils.getExtension(f);
        if (extension != null) {
            if (extension.equals(Utils.tiff) ||
                extension.equals(Utils.tif) ||
                extension.equals(Utils.gif) ||
                extension.equals(Utils.jpeg) ||
                extension.equals(Utils.jpg) ||
                extension.equals(Utils.png)) {
                return true;
            } else {
                return false;
            }
        }

        return false;
    }
    public String getDescription() {
        return "Just Images";
    }
}
```

3.5.3 Επεξήγηση της κλάσης ImagePreview.

```
import javax.swing.*;
import java.beans.*;
import java.awt.*;
import java.io.File;
//Εδώ έχουμε την δημιουργία της προεπισκόπησης των εικόνων.
public class ImagePreview extends JComponent
    implements PropertyChangeListener {
    ImageIcon thumbnail = null;
    File file = null;
//ορισμός προτιμώμενων διαστάσεων
```

```

public ImagePreview(JFileChooser chooser) {
    setPreferredSize(new Dimension(100, 50));
    chooser.addPropertyChangeListener(this);
}
public void loadImage() {
    if (file == null) {
        thumbnail = null;
        return;
    }
    ImageIcon tmpIcon = new ImageIcon(file.getPath());
    if (tmpIcon != null) {
        if (tmpIcon.getIconWidth() > 90) {
            thumbnail = new ImageIcon(tmpIcon.getImage().
                getScaledInstance(90, -1,
                    Image.SCALE_DEFAULT));
        } else { //no need to miniaturize
            thumbnail = tmpIcon;
        }
    }
}

public void propertyChange(PropertyChangeEvent e) {
    boolean update = false;
    String prop = e.getPropertyName();
    if (JFileChooser.DIRECTORY_CHANGED_PROPERTY.equals(prop)) {
        file = null;
        update = true;
        //εδώ γίνεται η εισάγεται το αρχείο που έχει επιλεγθεί.
    } else if (JFileChooser.SELECTED_FILE_CHANGED_PROPERTY.equals(prop)) {
        file = (File) e.getNewValue();
        update = true;
    }

    // ανανέωση της προεπισκόπησης
    if (update) {
        thumbnail = null;
        if (isShowing()) {
            loadImage();
            repaint();
        }
    }
}

protected void paintComponent(Graphics g) {
    if (thumbnail == null) {
        loadImage();
    }
}

```

```

if (thumbnail != null) {
    int x = getWidth()/2 - thumbnail.getIconWidth()/2;
    int y = getHeight()/2 - thumbnail.getIconHeight()/2;

    if (y < 0) {
        y = 0;
    }

    if (x < 5) {
        x = 5;
    }
    thumbnail.paintIcon(this, g, x, y);
}
}
}

```

3.5.4 Επεξήγηση της κλάσης Utils.

Την Utils χρησιμοποιεί η ImageFilter.

```

import java.io.File;
import javax.swing.ImageIcon;

public class Utils {
    public final static String jpeg = "jpeg";
    public final static String jpg = "jpg";
    public final static String gif = "gif";
    public final static String tiff = "tiff";
    public final static String tif = "tif";
    public final static String png = "png";
    // εδώ δέχεται την κατάληξη του αρχείου
    public static String getExtension(File f) {
        String ext = null;
        String s = f.getName();
        int i = s.lastIndexOf('.');

        if (i > 0 && i < s.length() - 1) {
            ext = s.substring(i+1).toLowerCase();
        }
        return ext;
    }

    //Επιστρέφει ένα ImageIcon, ή null στην περίπτωση που το path δεν είναι έγκυρο.

    protected static ImageIcon createImageIcon(String path) {
        java.net.URL imgURL = Utils.class.getResource(path);
        if (imgURL != null) {

```

```

        return new ImageIcon(imgURL);
    } else {
        System.err.println("Couldn't find file: " + path);
        return null;
    }
}
}
}

```

3.6 Επεξήγηση των ActionListeners των κουμπιών επιλογής χρώματος (JColorChoosers)

Ο παρακάτω κώδικας υπάρχει σε κάθε κουμπί επιλογής χρώματος της εφαρμογής.

```

private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
    JColorChooser chooser=new JColorChooser();
    Color c;
    c=JColorChooser.showDialog(this,"door main color",getBackground());
    jButton38.setBackground(c);
}

```

3.7 Επεξήγηση της κλάσης RoomObjects

Για να πραγματοποιηθεί το update των αντικειμένων του δωματίου , δημιουργήθηκε η κλάση RoomObjects . Κάθε αντικείμενο στο δωμάτιο αποτελεί ένα αντικείμενο τύπου RoomObjects και ανάλογα με την κατηγορία στην οποία ανήκει τοποθετείται σε Vector.

```

import java.awt.*;
public class RoomObjects {

    public String name,type;
    public String material;
    public String style;
    public Color colorA,colorB;
    public String materialA;
    public String materialB;

    public RoomObjects()
    { }

    public RoomObjects(String name, String type,String materialA,String
materialB,String style,Color colorA,Color colorB)
    { this.name=name;
      this.type=type;

```

```

this.materialA=materialA;
this.materialB=materialB;
this.style=style;
this.colorA=colorA;
this.colorB=colorB;}
}

```

3.8 Επεξήγηση του ActionListener των κουμπιών Add των αντικειμένων του δωματίου.

3.8.1 ActionListener κουμπιού Add για τα αντικείμενα του Vector furniture.

// Ο παρακάτω κώδικας είναι ίδιος για όλα τα κουμπιά Add των αντικειμένων του Vector furniture.

```

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    boolean p=true;
    String previous=(String)jTextArea1.getText();
    if(jComboBox5.getSelectedItem()==null || jComboBox7.getSelectedItem()==null
        || jComboBox13.getSelectedItem()==null ||
jComboBox8.getSelectedItem()==null || jButton3.getBackground()==initial
        || jButton7.getBackground()==initial)
    { p=false;
      JOptionPane.showMessageDialog(this,"Please fill in all fields!");
    }

    else if (p=true){

        RoomObjects Sit = new RoomObjects();
        String ii=Integer.toString(i+1);
        Sit.name="Sit"+ii;
        Sit.type=(String)jComboBox5.getSelectedItem();
        Sit.materialA=(String)jComboBox7.getSelectedItem();
        Sit.style=(String)jComboBox13.getSelectedItem();
        Sit.materialB=(String)jComboBox8.getSelectedItem();
        Sit.colorA=jButton3.getBackground();
        Sit.colorB=jButton7.getBackground();

        furniture.addElement(Sit);
        jTextArea1.setText(previous+(\n')+Sit.type);
        i++;
    }
}
}

```


3.8.2 ActionListener κουμπιού Add για τα αντικείμενα του Vector accessories.

// Ο παρακάτω κώδικας είναι ίδιος για όλα τα κουμπιά Add του Vector accessories.

```
private void jButton24ActionPerformed(java.awt.event.ActionEvent evt) {
    boolean p=true;
    String previous=(String)jTextArea2.getText();
    if(jComboBox29.getSelectedItem()==null || jComboBox11.getSelectedItem()==null ||
    jButton22.getBackground()==initial)
        { p=false;
        JOptionPane.showMessageDialog(this,"Please fill in style,color,material
field!");
        }

    else if (p=true){
        RoomObjects Acc = new RoomObjects();
        String zi=Integer.toString(z+1);
        Acc.name="floor lighting object"+zi;
        Acc.type="floor lighting object";

        Acc.materialA=(String)jComboBox11.getSelectedItem();
        Acc.style=(String)jComboBox29.getSelectedItem();
        Acc.materialB("Material Index Name");
        Acc.colorA=jButton22.getBackground();
        Acc.colorB=initial;
        accessories.addElement(Acc);
        jTextArea2.setText(previous+(\n')+Acc.type);
        z++;
        }
    }
```

3.9 Επεξήγηση μεθόδων doHex, checkMatId , checkStyleId, checkShapeId και checkFloorId

Αυτές οι μέθοδοι χρησιμοποιούνται στον κώδικα που ακολουθεί για την ολοκλήρωση του update. Είναι χρήσιμο να δούμε τι κάνουν ώστε να είναι σαφής η χρήση τους στον ActionListener του finish.

3.9.1 Η doHex.

Μετατρέπει ένα χρώμα σε δεκαεξαδικό αριθμό

```
String doHex(Color c1) {  
  
    String red = Integer.toHexString(c1.getRed());  
    String green = Integer.toHexString(c1.getGreen());  
    String blue = Integer.toHexString(c1.getBlue());  
    if (red.length() == 1)red = "0" + red;  
    if (green.length() == 1) green = "0" + green;  
    if (blue.length() == 1)blue = "0" + blue;  
    String hexc1 = "#" + red + green + blue;  
    return hexc1;  
}
```

3.9.2 checkMatId και checkStyleId

Έλεγχος υλικού και στυλ για απόδοση κωδικού.

Και οι δύο μέθοδοι λειτουργούν με τον ίδιο τρόπο. Παίρνουν την τιμή από το Combobox και ανατρέχουν το xml που βρίσκεται στο δίσκο. Μόλις βρουν την τιμή που ταιριάζει και για το πεδίο στο οποίο αναφέρεται, τότε τραβάνε τον κωδικό.

```
int checkMatId(String comboValue) {  
    int id=0; String id2=" ";  
    String patternStr;  
  
    try{  
        try{  
            try{  
                try{  
  
                    File f = new File("C:\\\\Configuration\\\\Combobox.xml");  
                    FileReader rd = new FileReader(f);  
                    char[] buf = new char[(int)f.length()];  
                    rd.read(buf);  
                    patternStr = new String(buf);  
                    InputStream is = new ByteArrayInputStream(patternStr.getBytes("utf-8"));  
                    DocumentBuilderFactory docBuilderFactory =  
                    DocumentBuilderFactory.newInstance();  
                    DocumentBuilder docBuilder =docBuilderFactory.newDocumentBuilder();
```

```

Document myXML =docBuilder.parse(is);
myXML.normalize();
Element myXMLElement = myXML.getDocumentElement();
    NodeList commonList =
myXMLElement.getElementsByTagName("Drop_Down");
    int ff=0;
    while (ff<commonList.getLength()){
        Node commonNode = commonList.item(ff);
        Element commonElement = (Element)commonNode;
        String b=(String)commonElement.getAttribute("name");
        if (b.equalsIgnoreCase("material")){
            NodeList nodeList1=commonElement.getElementsByTagName("Item");
            int k=0;
            while (k<nodeList1.getLength()){
                Node node1 = nodeList1.item(k);
                Element element1 = (Element)node1;
                Node myNode = node1.getNextSibling();
                myNode=element1.getFirstChild();
                String a =myNode.getNodeValue();
                if (comboValue.equalsIgnoreCase(a)){
                    id2 =element1.getAttribute("id");
                }
                k++;
            }
            ff++;
        }
        id= Integer.parseInt(id2);
    }
    catch (FileNotFoundException e){
    }
}

    catch (java.io.IOException e){
    }

    }

    catch (ParserConfigurationException e){
    }
    }

    catch (org.xml.sax.SAXException e){
    }
    }

    catch(Exception e){
    e.printStackTrace();}

```

```

        return id;
    }

int checkStyleId(String comboValue) {
    int id=0; String id2=" ";
    String patternStr;
    try{
        try{
            try{
                try{
                    File f = new File("C:\\Configuration\\Combobox.xml");
                    FileReader rd = new FileReader(f);
                    char[] buf = new char[(int)f.length()];
                    rd.read(buf);
                    patternStr = new String(buf);
                    InputStream is = new ByteArrayInputStream(patternStr.getBytes("utf-8"));
                    DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
                    DocumentBuilder docBuilder =docBuilderFactory.newDocumentBuilder();
                    Document myXML =docBuilder.parse(is);
                    myXML.normalize();
                    Element myXMLElement = myXML.getDocumentElement();
                    NodeList commonList =
myXMLElement.getElementsByTagName("Drop_Down");
                    int ff=0;
                    while (ff<commonList.getLength()){
                        Node commonNode = commonList.item(ff);
                        Element commonElement = (Element)commonNode;
                        String b=(String)commonElement.getAttribute("name");
                        if (b.equalsIgnoreCase("style")){
                            NodeList nodeList1=commonElement.getElementsByTagName("Item");
                            int k=0;
                            while (k<nodeList1.getLength()){
                                Node node1 = nodeList1.item(k);
                                Element element1 = (Element)node1;
                                Node myNode = node1.getNextSibling();
                                myNode=element1.getFirstChild();
                                String a =myNode.getNodeValue();
                                if (comboValue.equalsIgnoreCase(a)){
                                    id2 =element1.getAttribute("id");
                                    System.out.println(id2);
                                }
                                k++;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```
    }
    ff++;
}
id= Integer.parseInt(id2);
System.out.println(id);
}
catch (FileNotFoundException e){
}
}

catch (java.io.IOException e){
}

}

catch (ParserConfigurationException e){
}
}

catch (org.xml.sax.SAXException e){
}
}
catch(Exception e){
e.printStackTrace();}

return id;
}
```

3.10 Επεξήγηση του ActionListener του κουμπιού finish.

Το κουμπί finish είναι το κουμπί με το πάτημα του οποίου γίνεται το update και τερματίζεται η λειτουργία του wizard.

Με το πάτημα του κουμπιού πραγματοποιούνται οι εξής λειτουργίες.

Μεταφορά ενός αντιγράφου του αρχείου μοντέλου από την exist στο σκληρό δίσκο. Στη συνέχεια το αντίγραφο μεταφέρεται και πάλι στην exist σε έναν διαφορετικό φάκελο , μετονομασμένο. Πανω στο αντίγραφο αυτό πραγματοποιούνται xquery updates , πρώτα από τις περιγραφικές επιλογές του wizard (με την κλήση της doUpdate), και στη συνέχεια xquery updates που αφορούν στα αντικείμενα του δωματίου (με την κλήση της doUpdatevectors).

```
//επιβεβαίωση από τον χρήστη ότι έχει τελειώσει με τις επιλογές και θέλει να προχωρήσει.
```

```
int n= JOptionPane.showConfirmDialog(null,"Are you sure you are done with your description and want to proceed?", "End of description confirmation", JOptionPane.YES_NO_OPTION);
```

```
if (n== JOptionPane.YES_OPTION) {
```

```
boolean pks=true;String patternStr;
```

```
//έλεγχος τιμών υποχρεωτικών πεδίων
```

```
if(jComboBox1.getSelectedItem()==null || jComboBox2.getSelectedItem()==null || jComboBox3.getSelectedItem()==null || jComboBox4.getSelectedItem()==null)
```

```
{ pks=false;
```

```
JOptionPane.showMessageDialog(this,"Please fill in room description!");
```

```
}
```

```
else
```

```
if(jComboBox15.getSelectedItem()==null || jComboBox16.getSelectedItem()==null || jButton6.getBackground()==initial)
```

```
{ pks=false;
```

```
JOptionPane.showMessageDialog(this,"Please fill in floor description!");
```

```
}
```

```
else if(jComboBox9.getSelectedItem()==null ||
```

```
jButton27.getBackground()==initial)
```

```
{ pks=false;
```

```
JOptionPane.showMessageDialog(this,"Please fill ceiling description!");
```

```
}
```

```
else if(jComboBox10.getSelectedItem()==null ||
```

```
jButton31.getBackground()==initial)
```

```
{ pks=false;
```

```
JOptionPane.showMessageDialog(this,"Please fill wall description!");
```

```

}else pks=true;

//παίρνουμε ημερομηνία και ώρα η οποία θα συνοδεύει το όνομα του τελικού
αρχείου.

if(pks==true){
    DateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy_HH.mm.ss");
    Date date = new Date();
    String currentDate=dateFormat.format(date);

//Δημιουργούμε σύνδεση με την Exist μέσω κατάλληλου driver και αποθηκεύουμε
στην μεταβλητή τύπου Collection col την collection που θα χρησιμοποιήσουμε.
Document doc2;
    Resource r = null;
    {
        FileReader rdnewXML = null;
        try {
            String driver = "org.exist.xmldb.DatabasImpl";
            Class cl = Class.forName(driver);
            Database database = (Database) cl.newInstance();
            DatabaseManager.registerDatabase(database);
            Collection col =
DatabaseManager.getCollection("xmldb:exist://localhost:8080/exist/xmlrpc/db/XMLr
oom");
            XPathQueryService service = (XPathQueryService)
col.getService("XPathQueryService", "1.0");
            service.setProperty("indent", "yes");

//Με ένα query παίρνουμε όλο το xml που θα χρησιμεύσει ως πρότυπο.
            ResourceSet result = service.query("for $x in doc('XML_29_4_09.xml')
return $x");
            ResourceIterator a= result.getIterator();
            while (a.hasMoreResources()) {
                r = a.nextResource();
            }

            Collection colfinal =
DatabaseManager.getCollection("xmldb:exist://localhost:8080/exist/xmlrpc/db/final
", "admin", "admin");

//Μέσω ενός XMLSerializer αποθηκεύουμε το περιεχόμενο του XML_29_4_09..xml
στο δίσκο μας. Επίσης δημιουργούμε σε καινούργιο collection στην Exist ένα
καινούργιο Document που είναι τύπου XMLResource και εκεί μέσα αποθηκεύουμε
το παρεχόμενο που πήραμε από το XML_29_4_09..xml

XMLResource documentbef = (XMLResource) r;

```

//Χρησιμοποιούμε XMLSerializer για να αποθηκεύσουμε το περιεχόμενο του XML_29_4_09.xml στο δίσκο μας .

```
doc2 = (Document) documentbef.getContentAsDOM();
doc2.normalize();
OutputFormat format = new OutputFormat(doc2,"UTF-8",true);
XMLSerializer serializer = new XMLSerializer(new FileOutputStream(new
File("C:\\Configuration\\XmlRoomCopy.xml")), format);

serializer.serialize(doc2);
String FinalOnoma = "XMLroom" + "_" + currentDate;
XMLResource document = (XMLResource)
colfinal.createResource(FinalOnoma,"XMLResource");
XPathQueryService servicefinal = (XPathQueryService)
colfinal.getService("XPathQueryService", "1.0");
servicefinal.setProperty("indent", "yes");
File f2 = new File("C:\\Configuration\\XmlRoomCopy.xml");
if (!f2.canRead()) {
    System.out.println("cannot read file");
    return;
}
document.setContent(f2);
System.out.println(document.getId());
colfinal.storeResource(document);

File newXML = new File("C:\\Configuration\\XmlRoomCopy.xml");
rdnewXML = new FileReader(newXML);
char[] bufminor = new char[(int) newXML.length()];
rdnewXML.read(bufminor);
patternStr = new String(bufminor);
```

//Περνάμε τα δεδομένα σε μια μεταβλητή τύπου InputStream και δημιουργούμε ένα αντικείμενο τύπου DocumentBuilderFactory. Μέσω αυτού δημιουργούμε ένα αντικείμενο τύπου docBuilder και στη συνέχεια κάνουμε parse τα δεδομένα μας σε ένα αντικείμενο τύπου Document και με όνομα XMLRoomCopy.xml.

```
InputStream isnewXML = new
ByteArrayInputStream(patternStr.getBytes("utf-8"));
DocumentBuilderFactory docBuilderFactoryXML =
DocumentBuilderFactory.newInstance();
DocumentBuilder docBuilderXML =
docBuilderFactoryXML.newDocumentBuilder();
Document myXMLnew = docBuilderXML.parse(isnewXML);
myXMLnew.normalize();
Element myXMLElement = myXMLnew.getDocumentElement();
```


3.10.1 Επεξήγηση της μεθόδου doUpdate.

```
void doUpdate(String dbName){

    String minorfunctionality;
    String majorfunctionality;
    String roomstyle,windowmaterial,windowcolor;
    String roommood,doortype,doormaterial,doorcolor;
    String floormaterial,floorcolor,floorstyle;
    String ceilingmaterial,wallmaterial,wallcolor;
    String ceilingcolor,lightsource;
    String utilities;

    //χρήση μεθόδων για απόδοση (ή μετατροπή των τιμών στην κατάλληλη μορφή)
    //στις μεταβλητές που θα χρησιμοποιηθούν στα queries.
    minorfunctionality = (String) jComboBox2.getSelectedItemAt();
    majorfunctionality = (String) jComboBox1.getSelectedItemAt();
    roomstyle = (String) jComboBox3.getSelectedItemAt();
    roommood = (String) jComboBox4.getSelectedItemAt();
    ceilingmaterial = (String) jComboBox9.getSelectedItemAt();
    floormaterial = (String) jComboBox15.getSelectedItemAt();
    ceilingcolor= doHex(jButton27.getBackground());
    doortype=(String)jComboBox32.getSelectedItemAt();
    doormaterial=(String)jComboBox34.getSelectedItemAt();
    int ceilingmatcode=checkMatId(ceilingmaterial);
    wallmaterial= (String)jComboBox10.getSelectedItemAt();
    int wallmatcode= checkMatId(wallmaterial);
    int floormatcode= checkMatId(floormaterial);
    floorcolor= doHex(jButton6.getBackground());
    floorstyle=(String)jComboBox16.getSelectedItemAt();
    int floorstylecode= checkStyleId(floorstyle);
    int roomstylecode= checkStyleId(roomstyle);
    wallcolor= doHex(jButton31.getBackground());
    windowmaterial=(String)jComboBox36.getSelectedItemAt();

    // σύνδεση με την exist
    try{
        String driver = "org.exist.xmlldb.DatabaseImpl";
        Class c2 = Class.forName(driver);
        Database database = (Database) c2.newInstance();
        DatabaseManager.registerDatabase(database);
        Collection col2 =
        DatabaseManager.getCollection("xmlldb:exist://localhost:8080/exist/xmlrpc/db/final
", "admin", "admin");
        XPathQueryService service = (XPathQueryService)
col2.getService("XPathQueryService", "1.0");
```

```
service.setProperty("indent", "yes");
```

// Εκτέλεση των πρώτων queries που αφορούν το δωμάτιο

```
ResourceSet result2 = service.query(" update replace
doc('+dbName+')//functionality/MinorFunctionality with <MinorFunctionality
rankNo='a number that shows the priority in minor functionalities'>" +
minorfunctionality + "</MinorFunctionality>");
ResourceSet result3 = service.query(" update replace
doc('+dbName+')//functionality/MajorFunctionality with <MajorFunctionality>" +
majorfunctionality + "</MajorFunctionality>");
ResourceSet result4 = service.query(" update replace
doc('+dbName+')//room/style with <style styleIndexName='"+roomstyle+"
styleIndexURL='URL of XML of style Index'>" + roomstylecode + "</style>");
ResourceSet result5 = service.query(" update insert attribute
MoodCharacterization{'"+roommood+"} into doc('+dbName+')//room ");
//Int floortypeid = checkfloorId(flootype);
ResourceSet result6 = service.query(" update replace
doc('+dbName+')//ceiling/ColorPalette with <ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>"+ceilingcolor+"</ColorPalette>");
ResourceSet result70 = service.query(" update replace
doc('+dbName+')//ceiling/material/materialType with <material><materialType
materialIndexName='"+ceilingmaterial+" materialIndexURL='URL of Material
Index'>"+ceilingmatcode+"</materialType><pattern patternIndexName='style index
name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material>");
```

//Κάποια πεδία του οδηγού, δεν είναι υποχρεωτικά. Μπορεί δηλαδή να συμπληρωθούν , μπορεί όμως και όχι. Γίνεται κάποιος έλεγχος των αντιστοιχων πεδίων που τα αφορούν για να διαπιστωθεί εάν θα γίνει το update.

```
if (jComboBox31.getSelectedItem()!=null && jComboBox12.getSelectedItem()!=null
&& jButton30.getBackground()!=initial)
{ lightsource= doHex(jButton30.getBackground());
ResourceSet result71 = service.query(" update replace
doc('+dbName+')//ceiling/lightSource/lightSourcePalletIndex with
<lightSourcePalletIndex lightSourcePalletIndexName='light Pallet index name'
lightSourcePalletIndexURL='URL of XML of style
Index'>"+lightsource+"</lightSourcePalletIndex>");}

ResourceSet result72 = service.query(" update replace
doc('+dbName+')//wall/material/materialType with <materialType
materialIndexName='"+wallmaterial+" materialIndexURL='URL of Material
Index'>"+wallmatcode+"</materialType>");
ResourceSet result73 = service.query(" update replace
```

```

doc(""+dbName+"")//wall/ColorPalette with <ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'+wallcolor+"</ColorPalette>");

    if (jComboBox32.getSelectedItem()!=null &&
jComboBox34.getSelectedItem()!=null && jButton38.getBackground()!=initial)
    {
doorcolor= doHex(jButton38.getBackground());int doormatcode=
checkMatId(doormaterial);

    ResourceSet result74 = service.query(" update replace
doc(""+dbName+"")//wall/door/material/materialType with <materialType
materialIndexName='"+doormaterial+"' materialIndexURL='URL of Material
Index'+doormatcode+"</materialType>");
    //ResourceSet result5 = service.query(" update replace attribute type with
type{'+doortype+'} into doc(""+dbName+"")//wall/door ");
    ResourceSet result75 = service.query(" update replace
doc(""+dbName+"")//wall/door/frame/ColorPalette with <ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'+doorcolor+"</ColorPalette>");
    ResourceSet result81 = service.query(" update insert attribute
type{'+doortype+'} into doc(""+dbName+"")//wall/door ");}

    if (jComboBox36.getSelectedItem()!=null &&
jButton40.getBackground()!=initial)
    {
    windowcolor= doHex(jButton40.getBackground());int
windowmatcode=checkMatId(windowmaterial);
    ResourceSet result76 = service.query(" update replace
doc(""+dbName+"")//wall/window/material/materialType with <materialType
materialIndexName='"+windowmaterial+"' materialIndexURL='URL of Material
Index'+windowmatcode+"</materialType>");
    ResourceSet result77 = service.query(" update replace
doc(""+dbName+"")//wall/window/frame/ColorPalette with <ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'+windowcolor+"</ColorPalette>");
    // ResourceSet result81 = service.query(" update insert attribute
type{'+windowtype+'} into doc(""+dbName+"")//wall/window ");}
    }

    ResourceSet result78= service.query(" update replace
doc(""+dbName+"")//floor/material/materialType with <materialType
materialIndexName='"+floormaterial+"' materialIndexURL='URL of Material
Index'+floormatcode+"</materialType>");
    ResourceSet result79 = service.query(" update replace
doc(""+dbName+"")//floor/ColorPalette with <ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette

```

```
Index'>"+floorcolor+"</ColorPalette>");
    ResourceSet result80 = service.query(" update replace
doc(""+dbName+"")//floor/style with <style styleIndexName='"+floorstyle+"'
styleIndexURL='URL of XML of style Index'>" + floorstylecode + "</style>");
```

//Το πάνελ με τις επιλογές του χαλιού δεν έχει κουμπί add. Πρόκειται όμως για άλλο ένα αντικείμενο τύπου RoomObjects. Γίνεται λοιπόν ένας έλεγχος για την ύπαρξη του χαλιού. Εάν όλα τα πεδία έχουν κάποια τιμή τότε γίνεται η προσθήκη του αντικειμένου carpet στο Vector accessories. Συγκεκριμένα θα είναι πάντα το τελευταίο αντικείμενο μέσα στο Vector.

```
if (jComboBox28.getSelectedItem()!=null && jButton21.getBackground()!=initial
&& jButton23.getBackground()!=initial )
{ //carpet update
    RoomObjects carpet= new RoomObjects();
    carpet.type="carpet";
    carpet.name="carpet";
    carpet.style=(String)jComboBox28.getSelectedItem();
    carpet.colorA=jButton21.getBackground();
    carpet.colorB=jButton23.getBackground();
    carpet.materialA("Material Index Name");
    int carpetmatcode = checkMatId(carpet.materialA);
    int carpetstyle = checkStyleId(carpet.style);
    String carpetcolor = doHex(carpet.colorA);
    String carpetcolor2 =doHex(carpet.colorB);

    accessories.addElement(carpet);
    } catch (XMLDBException ex) {
        Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

3.10.2 Επεξήγηση της μεθόδου doUpdateVectors.

```
void doUpdateVectors(String dbName){

    try{
        String driver = "org.exist.xmlldb.DatabaseImpl";
        Class c5 = Class.forName(driver);
        Database database = (Database) c5.newInstance();
        DatabaseManager.registerDatabase(database);
        Collection col2 =
DatabaseManager.getCollection("xmlldb:exist://localhost:8080/exist/xmlrpc/db/final
", "admin", "admin");
        XPathQueryService service = (XPathQueryService)
col2.getService("XPathQueryService", "1.0");
        service.setProperty("indent", "yes");

// πρώτα σαρώνεται το Vector furniture, το οποίο περιλαμβάνει τα έπιπλα του
δωματίου. Αφού παίρνουμε τις τιμές και όπου είναι αναγκαίο τις μετατρέπουμε
στην κατάλληλη μορφή, πραγματοποιούνται τα queries.

        int i=0;
        while (i<furniture.size())
        {
            int matcode= checkMatId(furniture.get(i).materialA);
            int stylecode= checkStyleId(furniture.get(i).style);
            String color = doHex(furniture.get(i).colorA);
            String colorb=doHex(initial);int matcode2=0;
            if (furniture.get(i).colorB!=initial) colorb= doHex(furniture.get(i).colorB);
            if (furniture.get(i).materialB!=null) matcode2=
checkMatId(furniture.get(i).materialB);

// Για το πρώτο αντικείμενο του Vector δεν θα γίνει update insert,αλλά update
replace.

            if(i==0)
            {
                ResultSet resultx2 = service.query(" update replace
doc(""+dbName+"")//floor/furniture with <furniture
type="" +furniture.get(i).type+""><model3D><coordinates> something,
something</coordinates><path>url of the x3d
model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style
styleIndexName="" +furniture.get(i).style+"" styleIndexURL='URL of XML of style
Index'>"+stylecode+""</style><style styleIndexName='style index name'
```

```

styleIndexURL='URL of XML of style Index'>style
code</style><material><materialType
materialIndexName="" +furniture.get(i).materialA+" materialIndexURL='URL of
Material Index'>" +matcode+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>" +color+"</ColorPalette><frame><material><materialType
materialIndexName="" +furniture.get(i).materialB+" materialIndexURL='URL of
Material Index'>" +matcode2+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>" +colorb+"</ColorPalette></frame></furniture>");
    i++;
}

```

// Τα επόμενα αντικείμενα γίνονται insert.

```

else{
    String query=("<furniture
type="" +furniture.get(i).type+"><model3D><coordinates> something,
something</coordinates><path>url of the x3d
model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style
styleIndexName="" +furniture.get(i).style+" styleIndexURL='URL of XML of style
Index'>" +stylecode+"</style><style styleIndexName='style index name'
styleIndexURL='URL of XML of style Index'>style
code</style><material><materialType
materialIndexName="" +furniture.get(i).materialA+" materialIndexURL='URL of
Material Index'>" +matcode+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>" +color+"</ColorPalette><frame><material><materialType
materialIndexName="" +furniture.get(i).materialB+" materialIndexURL='URL of
Material Index'>" +matcode2+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>" +colorb+"</ColorPalette></frame></furniture>");
    ResourceSet resultx1 = service.query(" update insert (" +query+") following
doc("" +dbName+"")//floor/ColorPalette");
    i++;
}

```

```

}

// Μετά προχωράμε στον Vector με τα accessories.

i=0;
while (i<accessories.size())
{
    int matcode21= checkMatId(accessories.get(i).materialA);
    int stylecode2= checkStyleId(accessories.get(i).style);
    String color2 = doHex(accessories.get(i).colorA);
    String colorb2=doHex(initial);int matcode22=0;
    if (accessories.get(i).colorB!=initial) colorb2=
doHex(accessories.get(i).colorB);
    if (accessories.get(i).materialB!=null) matcode22=
checkMatId(accessories.get(i).materialB);

    if(i==0)
    {
        ResourceSet resultx2 = service.query(" update replace
doc(""+dbName+"")//floor/accessories with <accessories
type="+accessories.get(wii).type+"><model3D><coordinates> something,
something</coordinates><path>url of the x3d
model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style
styleIndexName="" +accessories.get(wii).style+" styleIndexURL='URL of XML of style
Index'" +stylecode2+"</style><material><materialType
materialIndexName="" +accessories.get(wii).materialA+" materialIndexURL='URL of
Material Index'" +matcode21+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'" +color2+"</ColorPalette><frame><material><materialType
materialIndexName="" +accessories.get(wii).materialB+" materialIndexURL='URL of
Material Index'" +matcode22+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'" +colorb2+"</ColorPalette></frame></accessories>");
        System.out.println(i+accessories.get(i).type);
        i++;
    }
    else{
        String query=(" <accessories
type="+accessories.get(i).type+"><model3D><coordinates> something,
something</coordinates><path>url of the x3d

```



```

model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style
styleIndexName="" +accessories.get(i).style+"" styleIndexURL='URL of XML of style
Index'>"+stylecode2+"</style><material><materialType
materialIndexName="" +accessories.get(i).materialA+"" materialIndexURL='URL of
Material Index'>"+matcode21+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>"+color2+"</ColorPalette><frame><material><materialType
materialIndexName="" +accessories.get(i).materialB+"" materialIndexURL='URL of
Material Index'>"+matcode22+"</materialType><pattern patternIndexName='style
index name' patternIndexURL='URL of XML of pattern Index'>pattern index ID that
expresses the details shapes</pattern></material><ColorPalette
ColorPaletteIndexName='palette index name' ColorPaletteIndexURL='URL of palette
Index'>"+colorb2+"</ColorPalette></frame></accessories>");
    ResourceSet resultx1 = service.query(" update insert (" +query+" ) preceding
doc(""+dbName+"")//floor/Construction");
    System.out.println(i+accessories.get(i).type);
    i++;
    }
}

```

// Το ίδιο γίνεται και με τον τελευταίο Vector, wallaccessories.

```

i=0;
for (i=0;i<wallaccessories.size();i++)
{
    int matcode21= checkMatId(wallaccessories.get(i).materialA);
    int stylecode2= checkStyleId(wallaccessories.get(i).style);
    String color2 = doHex(wallaccessories.get(i).colorA);
    String colorb2=doHex(initial);int matcode22=0;
    if (wallaccessories.get(i).colorB!=initial) colorb2=
doHex(wallaccessories.get(i).colorB);
    if (wallaccessories.get(i).materialB!=null) matcode22=
checkMatId(wallaccessories.get(i).materialB);
    if(i==0)
    {
        ResourceSet resultx88 = service.query(" update replace
doc(""+dbName+"")//wall/accessories with <accessories
type="" +wallaccessories.get(i).type+""><model3D><coordinates> something,
something</coordinates><path>url of the x3d
model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style

```

```

styleIndexName='Style Index Name' styleIndexURL='URL of XML of style
Index'>code</style><material><materialType materialIndexName='Material Index
name' materialIndexURL='URL of Material Index'>code</materialType><pattern
patternIndexName='style index name' patternIndexURL='URL of XML of pattern
Index'>pattern index ID that expresses the details
shapes</pattern></material><ColorPalette ColorPaletteIndexName='palette index
name' ColorPaletteIndexURL='URL of palette
Index'>code</ColorPalette><frame><material><materialType
materialIndexName='Material index name' materialIndexURL='URL of Material
Index'>code</materialType><pattern patternIndexName='style index name'
patternIndexURL='URL of XML of pattern Index'>pattern index ID that expresses the
details shapes</pattern></material><ColorPalette ColorPaletteIndexName='palette
index name' ColorPaletteIndexURL='URL of palette
Index'>code</ColorPalette></frame></accessories>");
    System.out.println(i); //i++;
}

else{
    String query("<accessories
type='"+wallaccessories.get(i).type+"'><model3D><coordinates> something,
something</coordinates><path>url of the x3d
model3D</path></model3D><shape>SVG
something</shape><dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions><RelativePosition ReferenceObject='another object'>left right
above below across infront behind center top bottom abut</RelativePosition><style
styleIndexName='Style Index Name' styleIndexURL='URL of XML of style
Index'>code</style><material><materialType materialIndexName='Material Index
Name' materialIndexURL='URL of Material Index'>code</materialType><pattern
patternIndexName='style index name' patternIndexURL='URL of XML of pattern
Index'>pattern index ID that expresses the details
shapes</pattern></material><ColorPalette ColorPaletteIndexName='palette index
name' ColorPaletteIndexURL='URL of palette
Index'>code</ColorPalette><frame><material><materialType
materialIndexName='Material Index Name' materialIndexURL='URL of Material
Index'>code</materialType><pattern patternIndexName='style index name'
patternIndexURL='URL of XML of pattern Index'>pattern index ID that expresses the
details shapes</pattern></material><ColorPalette ColorPaletteIndexName='palette
index name' ColorPaletteIndexURL='URL of palette
Index'>code</ColorPalette></frame></accessories>");
    ResourceSet resultx89 = service.query(" update insert (" +query+") following
doc(" +dbName+"")//wall/Construction");
}
}

} catch (XMLDBException ex) {

```

```
    Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);  
} catch (InstantiationException ex) {  
    Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);  
} catch (IllegalAccessException ex) {  
    Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);  
} catch (ClassNotFoundException ex) {  
    Logger.getLogger(Wizard1.class.getName()).log(Level.SEVERE, null, ex);  
}
```

```
}
```

Κεφάλαιο 4. Παράδειγμα χρήσης της εφαρμογής

Θα εκτελέσουμε το wizard για να την περιγραφή του δωματίου της φωτογραφίας που ακολουθεί.



4.1 Το αρχείο Template (XML)

```
<xml>
<header>
  <File>filepath</File>
    <Date/>
    <Designer>"</Designer>
    <Company>'bhj'</Company>
    <Project>
<projectID>"</projectID>
      <projectName>"</projectName>
      <projectAddress>"</projectAddress>
      <projectDescription>"</projectDescription>
    </Project>
  <config>
<units>
  <distance>'m,ft,cm,mm,inch'</distance>
    <angle>degree</angle>
    <area>'m2...'</area>
    <volume>'m3'</volume>
  </units>
</config>
</header>
<room MoodCharacterization="mood">
  <model3D>
```

```

<coordinates>x,y,z virtual world</coordinates>
  <path>url of the x3d model3D</path>
</model3D>
<shape>SVG something</shape>
<dimension>number,number,number-|x|,|y|,|z| bounding box</dimension>
<style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
  <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
  <functionality>
    <MajorFunctionality> something perhaps we have to produce a certain enumeration (like a list box)inside
the schema</MajorFunctionality>
    <MinorFunctionality rankNo="a number that shows the priority in minor
functionalities"> something perhaps we have to produce a certain enumeration (like a list box)inside the
schema</MinorFunctionality>
  </functionality>
  <RoomCoveringFactor>percentage of the covered space (by furnitures etc)bounding box of
the room - Sum(bounding boxes of the furnitures)/bounding box of the room </RoomCoveringFactor>
  <lighting>in lux the light volume of the room (artificial and daylight as well)</lighting>
  <ceiling>
<model3D>
  <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <shape>SVG something</shape>
  <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
  <style styleIndexName="style index name" styleIndexURL="URL of XML of style
Index">style code</style>
    <material>
      <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of
XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
      <lightSource>
<model3D>
  <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <dimensions>number,number,number - |x|,|y|,|z| bounding box
dimensions</dimensions>
    <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
    <lightSourcePalletteIndex lightSourcePalletteIndexName="light Pallette index
name" lightSourcePalletteIndexURL="URL of XML of style Index">light source index
number</lightSourcePalletteIndex>
    </lightSource>
    <Construction>
<model3D>
  <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <dimensions>number,number,number - |x|,|y|,|z| bounding box
dimensions</dimensions>
    <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
    <material>
      <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>

```

```

        <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
        </Construction>
    </ceiling>
    <wall>
    <model3D>
        <coordinates> something, something</coordinates>
        <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above below across
infront behind center top bottom abut</RelativePosition>
    <material>
        <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of
XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
        <window type="window type">
        <model3D>
            <coordinates> something, something</coordinates>
            <path>url of the x3d model3D</path>
        </model3D>
        <shape>SVG something</shape>
        <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
            <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
            <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
        <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
            <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
            </material>
            <frame>
                <material>
                    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                    </material>
                    <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                    </frame>
                </window>
                <door type="door type">
                    <model3D>
                        <coordinates> something, something</coordinates>
                        <path>url of the x3d model3D</path>
                    </model3D>
                    <shape>SVG something</shape>
                    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
                        <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>

```

```

<style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
    <material>
        <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
            <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
            </material>
        <frame>
            <material>
                <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                    </material>
                    <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                </frame>
            </door>
        <Construction>
            <model3D>
                <coordinates> x,y,z virtual world</coordinates>
                    <path>url of the x3d model3D</path>
                </model3D>
                <dimensions>number,number,number - |x|,|y|,|z| bounding box
dimensions</dimensions>
                    <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
                    <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
                <material>
                    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                        <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                        </material>
                        <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                    </Construction>
                    <accessories type="accessory type">
                        <model3D>
                            <coordinates> something, something</coordinates>
                                <path>url of the x3d model3D</path>
                            </model3D>
                            <shape>SVG something</shape>
                            <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
                                <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
                                <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
                            <material>
                                <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                                    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                                    </material>
                                    <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                                </accessories>
                            </wall>
                        </floor>
                    </model3D>

```

```

<coordinates> something, something</coordinates>
    <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of style
Index">style code</style>
    <material>
    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of
XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
        <furniture type="furniture type">
        <model3D>
        <coordinates> something, something</coordinates>
        <path>url of the x3d model3D</path>
        </model3D>
        <shape>SVG something</shape>
        <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
        <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
        <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
        <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
        <material>
        <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
            <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
            </material>
            <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
            <frame>
            <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                </material>
                <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                </frame>
            </furniture>
            <accessories type="accessory type">
            <model3D>
            <coordinates> something, something</coordinates>
            <path>url of the x3d model3D</path>
            </model3D>
            <shape>SVG something</shape>
            <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
            <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
            <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
            <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>

```



```

        <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
        </accessories>
        <Construction>
    <model3D>
        <coordinates> x,y,z virtual world</coordinates>
            <path>url of the x3d model3D</path>
        </model3D>
        <dimensions>number,number,number - |x|,|y|,|z| bounding box
dimensions</dimensions>
            <RelativePosition ReferenceObject="another object">left right above
below across infront behind center top bottom abut</RelativePosition>
            <style styleIndexName="style index name" styleIndexURL="URL of XML of
style Index">style code</style>
            <material>
                <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                    </material>
                    <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                    </Construction>
                </floor>
                <stair type="stair type">
                    <model3D>
                        <coordinates> something, something</coordinates>
                            <path>url of the x3d model</path>
                        </model3D>
                        <shape>SVG something</shape>
                        <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
                        <RelativePosition ReferenceObject="another object">left right above below across
infront behind center top bottom abut</RelativePosition>
                        <style styleIndexName="style index name" styleIndexURL="URL of XML of style
Index">style code</style>
                        <material>
                            <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                                <pattern patternIndexName="style index name" patternIndexURL="URL of
XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                                </material>
                                <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                                <Railing>
                                    <material>
                                        <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                                            <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the details shapes</pattern>
                                            </material>
                                            <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
                                            </Railing>
                                        </stair>
                                    </room>
                                </xml>

```

4.2 Το αρχείο που προκύπτει μετά το τρέξιμο της εφαρμογής.

```

<xml>
  <header>
    <File>filepath</File>
    <Date/>
    <Designer>"</Designer>
    <Company>'bhj'</Company>
    <Project>
      <projectID>"</projectID>
      <projectName>"</projectName>
      <projectAddress>"</projectAddress>
      <projectDescription>"</projectDescription>
    </Project>
    <config>
      <units>
        <distance>'m,ft,cm,mm,inch'</distance>
        <angle>degree</angle>
        <area>'m2...'

```

```

<lightSource>
  <model3D>
    <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <dimensions>number,number,number - |x|,|y|,|z| bounding box dimensions</dimensions>
  <RelativePosition ReferenceObject="another object">
    left right above below across infront behind center top bottom abut

    </RelativePosition>
  <lightSourcePalletteIndex lightSourcePalletteIndexName="light Pallette index name"
lightSourcePalletteIndexURL="URL of XML of style Index">light source index number</lightSourcePalletteIndex>
  </lightSource>
  <Construction>
    <model3D>
      <coordinates> x,y,z virtual world</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <dimensions>number,number,number - |x|,|y|,|z| bounding box dimensions</dimensions>
    <RelativePosition ReferenceObject="another object">
      left right above below across infront behind center top bottom abut

      </RelativePosition>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
    <material>
      <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPalletteIndexName="palette index name" ColorPalletteIndexURL="URL of palette
Index">code</ColorPalette>
    </Construction>
  </ceiling>
  <wall>
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">
      left right above below across infront behind center top bottom abut

      </RelativePosition>
    <material>
      <materialType materialIndexName="Ciment" materialIndexURL="URL of Material
Index">9</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPalletteIndexName="palette index name" ColorPalletteIndexURL="URL of palette
Index">#ffffcc</ColorPalette>
      <window type="window type">
        <model3D>
          <coordinates> something, something</coordinates>
          <path>url of the x3d model3D</path>
        </model3D>
        <shape>SVG something</shape>
        <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
        <RelativePosition ReferenceObject="another object">
          left right above below across infront behind center top bottom abut

          </RelativePosition>
        <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style

```

```

code</style>
  <material>
    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
    <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
  </material>
  <frame>
    <material>
      <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
    </frame>
  </window>
  <door type="door type">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">
      left right above below across infront behind center top bottom abut
    </RelativePosition>
  </door>
  <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
  <material>
    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
    <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
  </material>
  <frame>
    <material>
      <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
    </frame>
  </door>
  <Construction>
    <model3D>
      <coordinates> x,y,z virtual world</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <dimensions>number,number,number - |x|,|y|,|z| bounding box dimensions</dimensions>
    <RelativePosition ReferenceObject="another object">
      left right above below across infront behind center top bottom abut
    </RelativePosition>
  </Construction>
  <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
  <material>
    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
    <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
  </material>

```

```

    <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
  </Construction>
  <accessories type="Photo frame">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
    <style styleIndexName="Style Index Name" styleIndexURL="URL of XML of style Index">code</style>
    <material>
      <materialType materialIndexName="Material Index Name" materialIndexURL="URL of Material
Index">code</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
    <frame>
      <material>
        <materialType materialIndexName="Material Index Name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
    </frame>
  </accessories>
  <accessories type="Photo frame">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
    <style styleIndexName="Style Index Name" styleIndexURL="URL of XML of style Index">code</style>
    <material>
      <materialType materialIndexName="Material Index name" materialIndexURL="URL of Material
Index">code</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
    <frame>
      <material>
        <materialType materialIndexName="Material index name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">code</ColorPalette>
    </frame>
  </accessories>
</wall>
<floor>
  <model3D>
    <coordinates> something, something</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>

```

```

</model3D>
<shape>SVG something</shape>
<dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
<style styleIndexName="Classic" styleIndexURL="URL of XML of style Index">0</style>
<material>
  <materialType materialIndexName="Tile" materialIndexURL="URL of Material
Index">0</materialType>
  <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
  </material>
  <ColorPalette ColorPalettelIndexName="palette index name" ColorPalettelIndexURL="URL of palette
Index">#ffffcc</ColorPalette>
  <furniture type="Bedside table">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
    <style styleIndexName="Modern" styleIndexURL="URL of XML of style Index">'I'</style>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
    <material>
      <materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">I</materialType>
      <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
      </material>
      <ColorPalette ColorPalettelIndexName="palette index name" ColorPalettelIndexURL="URL of palette
Index">#ffcc99</ColorPalette>
      <frame>
        <material>
          <materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">I</materialType>
          <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
          </material>
          <ColorPalette ColorPalettelIndexName="palette index name" ColorPalettelIndexURL="URL of
palette Index">#ffcc66</ColorPalette>
          </frame>
        </furniture>
        <furniture type="Bed">
          <model3D>
            <coordinates> something, something</coordinates>
            <path>url of the x3d model3D</path>
          </model3D>
          <shape>SVG something</shape>
          <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
          <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
          <style styleIndexName="Modern" styleIndexURL="URL of XML of style Index">'I'</style>
          <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
          <material>
            <materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">I</materialType>
            <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
            </material>
            <ColorPalette ColorPalettelIndexName="palette index name" ColorPalettelIndexURL="URL of palette
Index">#ffcc66</ColorPalette>
            <frame>
              <material>
                <materialType materialIndexName="Wood" materialIndexURL="URL of Material

```

```

Index">|</materialType>
    <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">#ffcc66</ColorPalette>
    </frame>
</furniture>
<furniture type="Bed">
    <model3D>
        <coordinates> something, something</coordinates>
        <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above below across infront behind
center top bottom abut</RelativePosition>
    <style styleIndexName="Modern" styleIndexURL="URL of XML of style Index">|</style>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
    <material>
        <materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">|</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">#ffcc66</ColorPalette>
        <frame>
            <material>
                <materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">|</materialType>
                <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
                </material>
                <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of
palette Index">#ffcc66</ColorPalette>
                </frame>
            </furniture>
            <accessories type="accessory type">
                <model3D>
                    <coordinates> something, something</coordinates>
                    <path>url of the x3d model3D</path>
                </model3D>
                <shape>SVG something</shape>
                <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
                <RelativePosition ReferenceObject="another object">
                    left right above below across infront behind center top bottom abut

                    </RelativePosition>
                <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
                <material>
                    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
                <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
                </material>
                <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
            </accessories>
            <Construction>
                <model3D>
                    <coordinates> x,y,z virtual world</coordinates>
                    <path>url of the x3d model3D</path>
                </model3D>

```

```

<dimensions>number,number,number - |x|,|y|,|z| bounding box dimensions</dimensions>
<RelativePosition ReferenceObject="another object">
    left right above below across infront behind center top bottom abut
</RelativePosition>
<style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
<material>
    <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
    <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
</material>
<ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
</Construction>
</floor>
<stair type="stair type">
    <model3D>
        <coordinates> something, something</coordinates>
        <path>url of the x3d model</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding box</dimensions>
    <RelativePosition ReferenceObject="another object">
        left right above below across infront behind center top bottom abut
    </RelativePosition>
    <style styleIndexName="style index name" styleIndexURL="URL of XML of style Index">style
code</style>
    <material>
        <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
        <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
    <Railing>
        <material>
            <materialType materialIndexName="material index name" materialIndexURL="URL of Material
Index">code</materialType>
            <pattern patternIndexName="style index name" patternIndexURL="URL of XML of pattern
Index">pattern index ID that expresses the details shapes</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="palette index name" ColorPaletteIndexURL="URL of palette
Index">code</ColorPalette>
    </Railing>
</stair>
</room>
</xml>

```


Κεφάλαιο 5. Παράρτημα

5.1 Xml

Η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο. Σχεδόν όλα τα έγγραφα έχουν την ίδια δομή.

Η XML είναι κάτι περισσότερο από markup language είναι metalanguage, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες. Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται στη διατύπωση και την εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML τα tags είναι προκαθορισμένα ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τα tags και τις δομημένες μεταξύ τους σχέσεις.

Τα XML έγγραφα δεν είναι πολύπλοκα αλλά απλά και πολύ αποτελεσματικά. Το διδακτικό υλικό της well-formed XML αναλύει τη δημιουργία των XML εγγράφων, η οποία είναι κατά κάποιο τρόπο ίδια με την HTML καθώς επιτρέπει τη μη δομημένη δημιουργία εγγράφου. Η valid XML είναι πιο σύνθετη. Απαιτεί την ύπαρξη ενός Document Type Definition πριν να γραφεί το έγγραφο αλλά παρέχει μια γενική δομή με βάση την οποία τη δημιουργούμε.

Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών (data objects) που καλούνται XML έγγραφα (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται.

Τα XML έγγραφα αποτελούνται από μονάδες αποθήκευσης που καλούνται *entities* (οντότητες), οι οποίες περιέχουν πληροφορίες αναλυμένες ή μη. Οι αναλυμένες πληροφορίες αποτελούνται από *χαρακτήρες* (*characters*) οι οποίοι συνθέτουν *character data* και άλλοι οι οποίοι συνθέτουν *markup*. Η μορφή markup κωδικοποιεί την περιγραφή της τελικής αποθήκευσης του εγγράφου καθώς και τη λογική δομή.

Παράδειγμα αρχείου xml:

```
<person>                                     Opening tag
<name language="en">Anna Wintour</name>      Attribute
<age>30</age>
<sex >female</sex>
</person>                                     Closing tag
```

Πίνακας 1 «Παράδειγμα αρχείου xml»

5.2 XPath

Η XPath είναι μια γλώσσα για αναζήτηση πληροφοριών σε ένα έγγραφο XML. Την XPath μπορούμε να την χρησιμοποιήσουμε για να έχουμε μια επιτυχή περιήγηση σε στοιχεία και χαρακτηριστικά ενός εγγράφου XML. Η XPath είναι ένα σημαντικό στοιχείο του W3C XSLT προτύπου. Δεν είναι τυχαίο πως και η XQuery και η XPointer βασίζονται σε εκφράσεις XPath. Έτσι η κατανόηση της XPath δίνει μεγάλα πλεονεκτήματα ως προς την χρησιμοποίηση της XML.

Παράδειγμα XPath:

```
doc("catalog.xml")//product
```

Πίνακας 2 «Παράδειγμα XPath»

5.3 XQuery

Η XQuery ή αλλιώς XML Query περιγράφει μια γλώσσα αναζήτησης βάσεων δεδομένων για δεδομένα XML. Ο καλύτερος τρόπος να εξηγήσεις τι είναι η XML είναι να πεις ότι είναι το αντίστοιχο της SQL σε έγγραφα XML. Η XQuery 1.0 δημιουργήθηκε από το XML Query working group του διεθνή οργανισμού W3C. Η ανάπτυξη της ήταν συνδεδεμένη με την ανάπτυξη της XSLT 2.0 από το XSL Working Group.

Τα δυο group είναι υπεύθυνα για την δημιουργία της XPath 2.0 που είναι υποσύνολο της XQuery 1.0. Η XQuery 1.0 έγινε W3C πρότυπο τον Ιανουάριο του 2007.

Παράδειγμα xQuery με FLWORS expression :

```
for $prod in doc("catalog.xml")/catalog/product
where $prod/@dept = "ACC"
order by $prod/name
return $prod/name
```

Πίνακας 3 «Παράδειγμα xQuery»

5.4 Exist

Η Exist-db είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης βάσεων δεδομένων που βασίζεται εξολοκλήρου στην τεχνολογία XML. Αποθηκεύει XML έγγραφα με βάση το XML data model.






Η Exist-db υποστηρίζει πολλά διαδικτυακά τεχνολογικά πρότυπα όπως:

- ✚ XQuery 1.0 / XPath 2.0
- ✚ XSLT 1.0 (χρησιμοποιώντας Apache Xalan) or XSLT 2.0 (προαιρετική χρησιμοποίηση Saxon)
- ✚ HTTP interfaces: REST, WebDAV, SOAP, XMLRPC, Atom Publishing Protocol
- ✚ XML database specific: XMLDB, XQJ/JSR-225, XUpdate, XQuery update extensions

Η Exist-db παρέχει ένα ισχυρό περιβάλλον για την ανάπτυξη διαδικτυακών εφαρμογών που βασίζονται στην XQuery και στα σχετικά πρότυπα.

Κεφάλαιο 6. Βιβλιογραφία

6.1 Πηγές

-  <http://www.w3schools.com/>
-  <http://www.exist-db.org>
-  http://en.wikipedia.org/wiki/Main_Page
-  <http://xqilla.sourceforge.net/XQueryUpdate>
-  <http://www.cafeconleche.org/books/xmljava/chapters/ch13.html>