

# Εργασία 2:

## Εργασία XSLT

Κουρουπέτρογλου Πραξιτέλης Νικόλαος

### Εισαγωγή

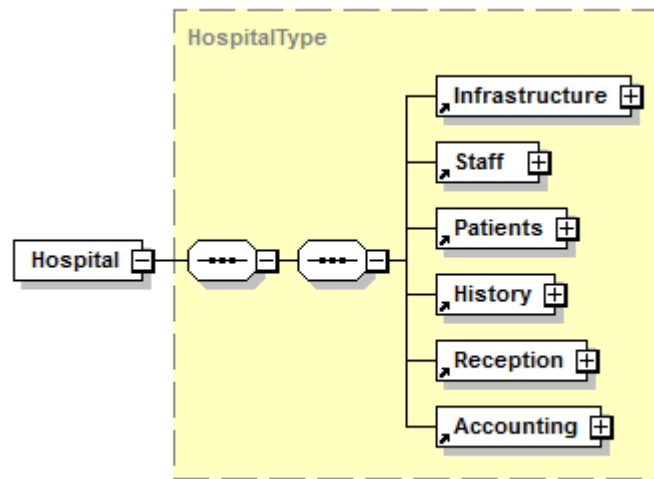
Η συγκεκριμένη εργασία ασχολήθηκε με τη δημιουργία XSLT αρχείων για τη μετατροπή από το αρχικό παραδοτέο του λεξιλογίου της XML για τη διαχείριση ενός νοσοκομείου σε αρχεία HTML. Η γενική ιδέα για το πώς λειτουργεί το συγκεκριμένο νοσοκομείο είναι η εξής:

- Το νοσοκομείο απαρτίζεται από Υποδομές Infrastructure που αυτές με τη σειρά του απαρτίζονται από πτέρυγες που περιέχουν τμήματα Departments του Νοσοκομείου (Τακτικά Ιατρεία, Καρδιολογική, Ογκολογική) και από τους Θαλάμους Wards νοσηλείας των ασθενών
- Το element Staff, που περιγράφει το ιατρικό και νοσηλευτικό προσωπικό.
- Το element Patients, που περιγράφει το ιατρικό και νοσηλευτικό προσωπικό.
- Το element Ιστορικά Ασθενών, που διατηρεί εγγραφές με την πορεία υγείας των ασθενών.
- Το element Διαχείριση Ραντεβού – Reception, που διατηρεί τα ραντεβού και του που πρέπει να μελλοντικά να εξεταστούν από γιατρούς
- Το element τμήμα Λογιστηρίου – Accounting, που διατηρεί τα έσοδα και τα έξοδα του Νοσοκομείου.

Συγκεκριμένα λοιπόν για τη αναπαράσταση των πληροφοριών που απαρτίζουν ένα πληροφορικό σύστημα Νοσοκομείου οι λειτουργικές μονάδες που απαρτίζουν ένα τέτοιο σύστημα είναι οι εξής:

- Υποδομές Νοσοκομείου - Infrastructure
- Προσωπικό - Staff
- Ασθενείς - Patients
- Ιστορικά Ασθενών - History
- Υποδοχή, Διαχείριση Ραντεβού - Reception
- Τμήμα Λογιστηρίου - Accounting

Η αρχική δομή ιεραρχικής αναπαράστασης είναι η εξής:

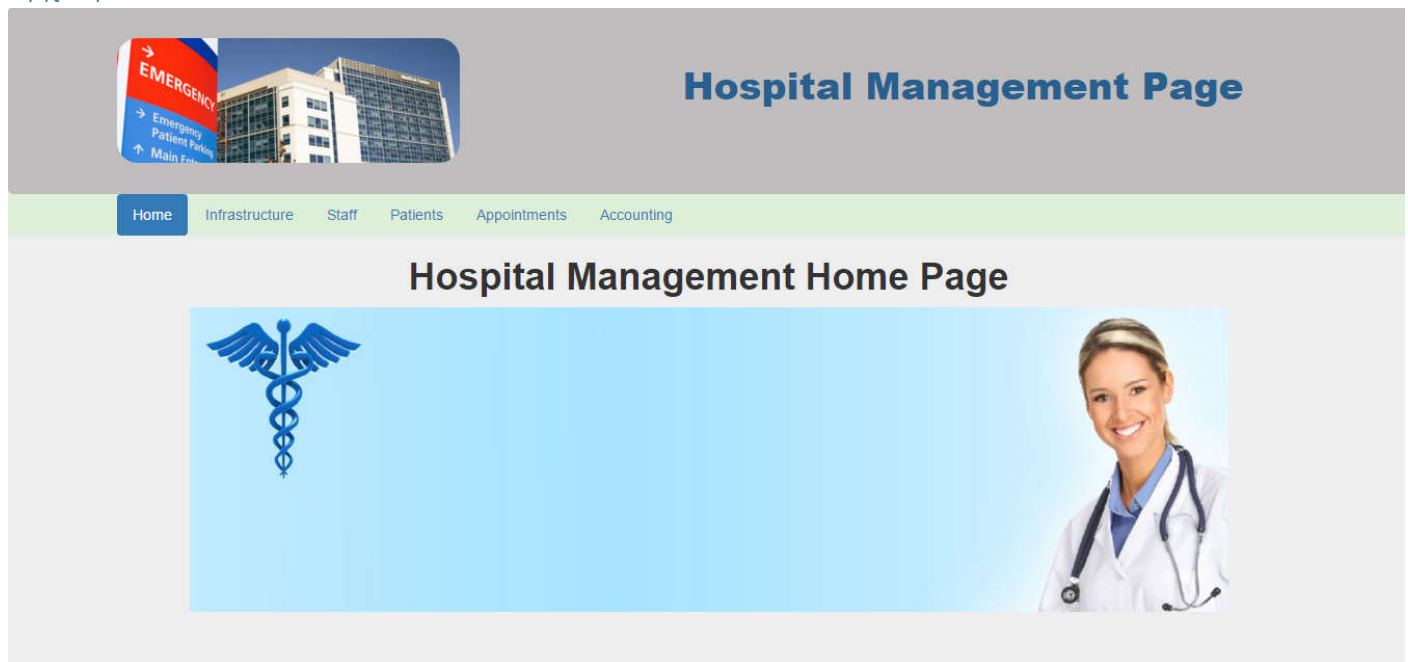


Πρέπει να σημειωθεί ότι δεν έγινε καμία αλλαγή πέρα από ελάχιστες στο XML Schema μας κάτι που δεν αλλάζει καθόλου το αρχικό παραδωτέο μας. Για αυτό το λόγο θα επισυναφθεί μαζί με τα υπόλοιπα παραδωτέα το προηγούμενο έγγραφο με τη παρουσίαση του Λεξιλογίου στην XML με τις λίγες αλλαγές που έγιναν.

### Λογισμικό Ανάπτυξης και Πορεία Ανάπτυξης XSLT

Το λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη της εργασίας και κυρίως των XSLT αρχείων είναι αυτό που προτάθηκε από το μάθημα. Για την ανάπτυξη και μετατροπή από XML σε HTML χρησιμοποιήθηκε η γλώσσα XSLT 2.0 αλλά και τεχνολογίες όπως CSS, Bootstrap, Javascript, JQuery UI ώστε να έχουμε μία εύχρηστη διασύνδεση με τη διεπαφή – σελίδα.

### Αρχική σελίδα



Η σελίδα μας ονομάζεται Hospital Management. Παρατηρούμε ότι έχουμε ένα μενού πλοήγησης για να πλοηγηθούμε στις διάφορες υποενότητες της σελίδας μας.



The screenshot shows a web application titled "Hospital Management Page". It features a navigation bar with links: Home, Infrastructure (selected), Staff, Patients, Appointments, and Accounting. The main content area is titled "Hospital Infrastructure - Basic Information". It displays a section for "Wing: Wing101" with an accordion showing "Departments in Wing Wing101" (Walk In Clinic, Nephrology, Microbiology) and "Wards in Wing Wing101" (Ward ID: Wing101Ward101). Below this, there are links for "Wing: Wing102" and "Wing: Wing103".

Το αρχείο Infrastructure.xslt μετασχηματίζει το αρχικό Hospital.xml αρχείο ώστε να παράγεται η σελίδα Infrastructure.html. Η συγκεκριμένη σελίδα παρουσιάζει τα περιεχόμενα της Υποδομής Infrastructure σε συνδιασμό με άλλα elements όπως το Προσωπικό Staff και οι ασθενείς. Χρησιμοποιήθηκε έντονα η χρήση του εργαλείου JQuery UI και της κλάσης του της accordion για να μπορούμε να ελέγχουμε την παραγόμενη πληροφορία σε τμήματα, να έχουμε καλύτερη πρόσβαση σε αυτή και μια καλύτερη ευχρηστία στην διεπαφή μας. Το παραγόμενο παραπάνω html αρχείο είναι αποτέλεσμα της χρήσης της Infrastructure.xslt πάνω στο αρχικό xml αρχείο μας. Τα περιεχόμενα της σελίδας απαρτίζονται από τις πτέρυγες του Νοσοκομείου όπου κάθε πτέρυγα αποτελείται από αντικείμενα τύπου Department – τμήματα και Κοιτώνες Wards. Τα Wards με τη σειρά τους αποτελούνται από αντικείμενα θαλάμους Rooms και αυτά κατ' επέκταση από κλίνες αντικείμενα τύπου Beds. Κάθε accordion περιέχει πληροφορίες για τα τμήματα του Νοσοκομείου, για τους κοιτώνες Wards που χρησιμοποιεί, για τις δεσμευμένες αλλά και για τις ελεύθερες και διαθέσιμες κλίνες που υπάρχουν για τους ασθενείς. Αυτό επιτεύχθηκε με τη χρήση αρκετών ερωτημάτων XPATH σε συνδιασμό με την XQUERY. Λόγο του όγκου του κάθε xslt αρχείου θα γίνει παρουσίαση της κάθε xslt δομής και της λογικής πίσω από τη χρήση της για τη μετατροπή από xml σε html. Ας δούμε τα πιο σημαντικά:

Αρχικά συνδέουμε το έγγραφό μας με τη ρίζα του δέντρου του αρχείου Hospital.xml αυτό γίνεται με την εξής εντολή:

```
<xsl:template match="/">
```

Στη συνέχεια χρησιμοποιούμε την παρακάτω xslt εντολή:

```
<xsl:for-each select="//Wing">
```

Κάνουμε χρήση της παραπάνω εντολής για να διασχίσουμε το xml αρχείο μας. Έτσι χρησιμοποιήθηκε η παραπάνω επαναληπτική δομή ώστε να έχουμε πρόσβαση σε όλες τα αντικείμενα τύπου Πτέρυγα ώστε στη συνέχεια να έχουμε πρόσβαση στις υποδομές (Departments και Wards). Το παραγόμενο html αποτέλεσμα είναι το εξής:

## Hospital Infrastructure - Basic Information

- Wing: Wing101
- Wing: Wing102
- Wing: Wing103

Η ίδια δομή χρησιμοποιήθηκε τόσο για τα Departments όσο και για Wards δηλαδή:

```
<xsl:for-each select="Departments/Department">
```

```
...
```

```
...
```

```
...
```

```
<xsl:for-each select="Wards/Ward">
```

Για κάθε τμήμα, προστέθηκαν διάφορες πληροφορίες που το χαρακτηρίζουν όπως το ID του, το όνομά του, το τηλέφωνό του κτλ. Αυτές έγιναν τόσο με τη διάσχιση της xml από την xslt αλλά και για την χρήση των δεδομένων της xml έγινε η χρήση της:

```
<xsl:value-of select=""/>
```

Και το επιθυμητό αποτέλεσμα:

### General Information about the Department

Department ID	Department Name	Department Phone
_Walk_in_Clinic	Walk In Clinic	202-555-0154

Πέρα από τις βασικές πληροφορίες για κάθε τμήμα, συνδυάστηκαν πληροφορίες και από άλλα τμήματα του xml δέντρου όπως το ιατρικό και νοσηλευτικό προσωπικό του. Αυτό επιτεύχθηκε με ερωτήματα XPATH στην XSLT. Για παράδειγμα το παρακάτω html:

### Employees Information

Total Number of Employees	Total Number of Doctors	Total Number of Nurses
8	4	4

Παράχθηκε με τα εξής ερωτήματα XPATH στην XSLT:

```
<xsl:variable name="works" select="@DeptartID"/>
```

```
<td><xsl:value-of
```

```
select="count(//Doctor[@WorksIn=$works]/DoctorName)+count(//Nurse[@WorksIn=$works]/NurseName)"/></td>
```

```
<td><xsl:value-of select="count(//Doctor[@WorksIn=$works]/DoctorName)"/></td>
```

```
<td><xsl:value-of select="count(//Nurse[@WorksIn=$works]/NurseName)"/></td>
```

Όπως βλέπουμε έγινε η χρήση της μεταβλητής xsl:variable για να διευκολύνει στα XPATH αιτήματά μας. Λόγο ότι έχουμε κάνει χρήση της `<xsl:for-each select="Departments/Department">` μπορούμε εύκολα να με προσωρινές μεταβλητές να παίρνουμε δεδομένα από τα αντικείμενα Department. Επίσης έγινε η χρήση της count της xslt για να μετρήσουμε τον αριθμό των στιγμιότυπων των εργαζομένων ανά περίπτωση του παραπάνω πίνακα.

Ένα άλλο τμήμα της html παράχθηκε και αξίζει να συζητηθεί είναι το παρακάτω τμήμα:

## Doctor Staff

Doctor Name	Speciality	Relation
Priscilla Arnold	Nephrologist	Trainee
Jenna Floyd	Nephrologist	Permanent
Ollie Holloway	Nephrologist	Visiting
Dennis Young	Nephrologist	Permanent

Που δημιουργήθηκε από το παρακάτω κομμάτι του κώδικα σε xslt:

```
<xsl:variable name="works" select="@DeptartID"/>
<xsl:for-each select="//Doctor[@WorksIn=$works]">
  <tbody>
    <tr>
      <td>
        <a href="Staff.html#{@DoctorID}" target="_blank"><xsl:value-of select="DoctorName"/></a>
      </td>
      <td>
        <xsl:value-of select="Speciality"/>
      </td>
      <td>
        <xsl:value-of select="Relantionship/@RelantionshipType"/>
      </td>
    </tr>
  </tbody>
</xsl:for-each>
```

Αρχικά αποθηκεύουμε σε μια προσωρινή μεταβλητή το id DeptartID, id του τμήματος που διασχίζουμε αυτή τη στιγμή. Με την for-each βρίσκουμε τους γιατρούς που εργάζονται σε αυτό το τμήμα με αυτό το id. Έπειτα για κάθε τέτοιο γιατρό δημιουργούμε ένα href link και κάνουμε χρήση του id του ως hashtag. Με αυτό τον τρόπο μπορώ να έχω πρόσβαση μέσω του συνδέσμου με το hashtag του σε άλλα κομμάτια του site Hospital Management.

Ένα άλλο ενδιαφέρον μέρος του html που παράγεται από τη μετατροπή της xml είναι το εξής:

Ward ID: Wing101Ward101

Used by Department: Nephrology  
 Is Inspected by the Nurse: Rachel Carter  
 Total Number of Rooms: 3  
 Total Number of Beds: 12  
 Available Beds: 11

Εκτός από τα Available Beds τα υπόλοιπα ακολουθούν παρώμοια λογική xslt προγραμματισμού. Θα εστιάσουμε στο τελευταίο που υποστηρίζεται από το παρακάτω xslt κώδικα:

```
<xsl:variable name="ward" select="@WardID"/></xsl:variable>
```

```

<xsl:for-each select="//Department">
  <xsl:if test="$ward = @useWard">
    <xsl:variable name="useWard" select="@useWard"></xsl:variable>
    <xsl:variable name="depart" select="@DepartID"></xsl:variable>
    <h4>Available Beds: <xsl:value-of select="count(/Ward[@WardID =
useWard]/Rooms/Room/Beds/Bed) - count(/Patient[@PatientType='InPatient']/AttendedDept[@AttendedIn=$depart])"></xsl:value-of></h4>
  </xsl:if>
</xsl:for-each>

```

Είναι ένα αρκετά πολύπλοκο ερώτημα XPATH και XSLT, ας το δούμε σιγά σιγά: στόχος μας είναι να βρούμε τον αριθμό των ελεύθερων κρεβατιών σε κοιτώνα Ward. Αρχικά αν και δεν φαίνεται στο σχήμα λόγο της μεγάλης έκτασης του κώδικα βρισκόμαστε σε ένα loop της μορφής: `<xsl:for-each select="Wards/Ward">`, που μας επιστρέφει όλους τους κόμβους Ward. Η προσωρινή μεταβλητή \$ward αποθηκεύει το τρέχον WardID. Στη συνέχεια τρέχουμε ένα `<xsl:for-each` για όλα τα department και παίρνουμε εκείνα μέσω της `xsl:if` που χρησιμοποιούν το τρέχον Ward που βρισκόμαστε. Έπειτα κρατάμε δύο προσωρινές μεταβλητές που μας βοηθούν στη τελευταία γραμμή του κώδικα που αφαιρούμε τα συνολικά κρεβάτια από το τρέχον Ward από εκείνα που χρησιμοποιούνται από ασθενείς που νοσηλεύονται εντός νοσοκομείου (InPatient) για το συγκεκριμένο Department που χρησιμοποιεί το τρέχον Ward κοιτώνα που βρισκόμαστε. Έτσι υπολογίζουμε τις ελεύθερες κλίνες.

Ένα τελευταίο κομμάτι της html είναι πως ξεχωρίζουμε τις ελεύθερες από τις κατειλημμένες κλίνες:

#### Beds in Room ID: Wing101Ward101Room101

Bed ID	Bed Size	Assigned To:
Wing101Ward101Room101Bed101	1	Angela Washington
Wing101Ward101Room101Bed102	2	Empty
Wing101Ward101Room101Bed103	1	Empty

Ο xslt κώδικα για το παραπάνω είναι η εξής τεχνική:

```

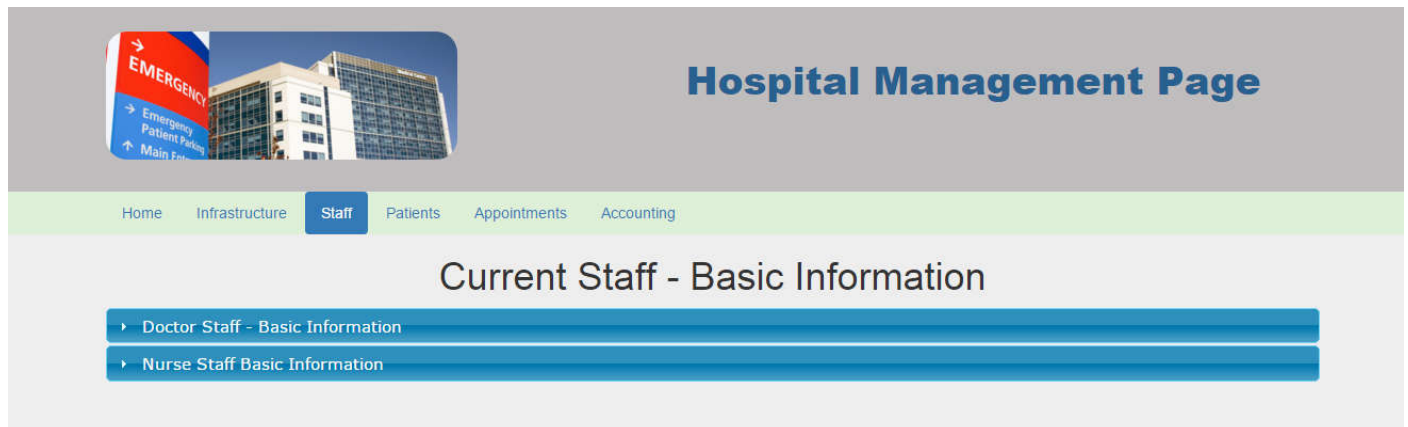
<xsl:choose>
  <xsl:when test="expression">
    ... some output ...
  </xsl:when>
  <xsl:otherwise>
    ... some output ....
  </xsl:otherwise>
</xsl:choose>

```

Χρησιμοποιήθηκε η συγκεκριμένη τεχνική ώστε χρωματίσουμε διαφορετικά τις κατειλημμένες κλίνες από τις ελεύθερες.



Το αρχείο Staff.xslt μετασχηματίζει το αρχικό μας αρχείο Hospital.xml στο αρχείο Staff.html. Σε αυτό περιγράφονται πληροφορίες για το ιατρικό και το νοσηλευτικό προσωπικό καθώς και στατιστικά στοιχεία για το καθένα από αυτά όπως φαίνεται στην παρακάτω εικόνα:



Η χρήση των δομών της xslt έγινε με το ίδιο τρόπο όπως και στη προηγούμενη σελίδα. Δηλαδή με τη χρήση `<xsl:for-each select="//Doctor">` και `<xsl:for-each select="//Nurse">` ας δούμε πως παράγονται τα στατιστικά στοιχεία για το σύνολο του ιατρικού προσωπικού όπως φαίνεται στο παρακάτω κομμάτι του παραγόμενου html αρχείου:

## Doctor Statistics

Total Number of Doctors: 21

Total Number of Permanent Doctors: 14

Total Number of Visiting Doctors: 3

Total Number of Trainee Doctors: 4

Και ο ανάλογο xslt κώδικας είναι ο εξής:

```
.....<h4>Total Number of Doctors: <xsl:value-of select="count(//Doctor)"></xsl:value-of></h4>
.....<h4>Total Number of Permanent Doctors: <xsl:value-of select="
count(//Doctor/Relationship[@RelationshipType='Permanent'])"></xsl:value-of></h4>
.....<h4>Total Number of Visiting Doctors: <xsl:value-of select="
count(//Doctor/Relationship[@RelationshipType='Visiting'])"></xsl:value-of></h4>
.....<h4>Total Number of Trainee Doctors: <xsl:value-of select="
count(//Doctor/Relationship[@RelationshipType='Trainee'])"></xsl:value-of></h4>
```

Πάλι και εδώ χρησιμοποιούμε τη συνάρτηση count της xslt και με τη χρήση φίλτρων παράγουμε το επιθυμητό αποτέλεσμα. Με την ίδια λογική παράγονται τα στατιστικά και για το νοσηλευτικό προσωπικό.

Ένα άλλο κομμάτι είναι οι πληροφορίες ανά γιατρό όπως παρουσιάζεται παρακάτω:

Doctor101							
Doctor ID	Doctor Name	Works In	Speciality	Relation	Address	Phones	Salary
Doctor101	Priscilla Arnold	Nephrology	Nephrologist	Trainee	326 Willow Street Rowlett, TX 75088	202-555-0168	1500 Salary101

Όλα τα στοιχεία ανά γιατρό τα προμηθευόμαστε από το αντικείμενο Doctor καθώς αυτό το διασχίζουμε με τη `<xsl:for-each select="//Doctor">`. Ενδιαφέρον παρουσιάζει το Salary με τον τρόπο που παράγεται:

```

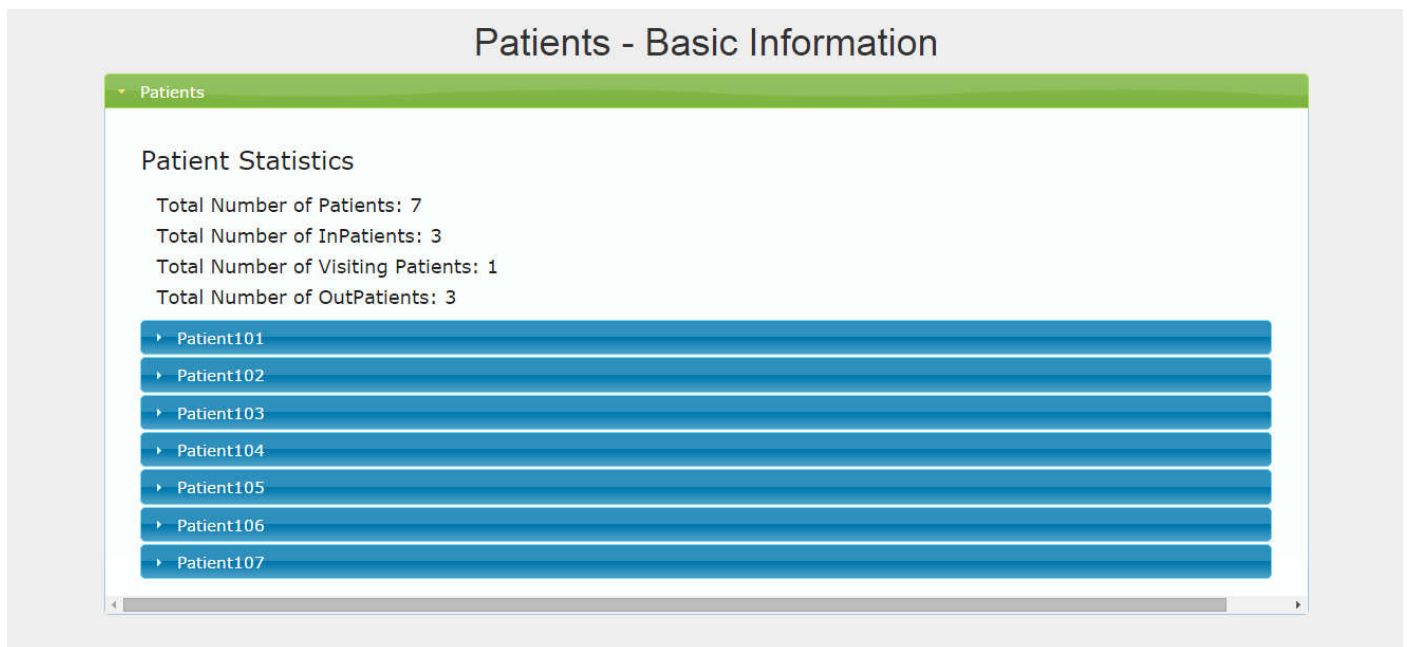
<td>
  <xsl:value-of select="//Salary[@staffID=$doctorID]/Payment/@amount"></xsl:value-of>
  <br></br>
  <a href="Accounting.html#//Salary[@staffID=$doctorID]/@SalaryID"><xsl:value-of select="
//Salary[@staffID=$doctorID]/@SalaryID"></xsl:value-of></a>
</td>

```

Πάλι εδώ χρησιμοποιούμε τα φίλτρα της XSLT και της XPATH για να επιστρέψουμε το μισθό του γιατρού αλλά και ένα σύνδεσμο στη σελίδα Accounting.html με hashtag το id που περιγράφει το μισθό του συγκεκριμένου γιατρού ώστε να έχουμε πρόσβαση σε συγκεκριμένες πληροφορίες στη σελίδα Accounting.html που θα δούμε παρακάτω.

### Patient.xslt - Patient.html

Το αρχείο Staff.xslt μετασχηματίζει το αρχικό μας αρχείο Hospital.xml στο αρχείο Patient.html. Σε αυτό περιγράφονται πληροφορίες για τους ασθενείς μας καθώς και στατιστικά στοιχεία για το σύνολό τους. Από το δέντρο Hospital.xml χρησιμοποιούμε σε συνδυασμό δύο βασικά αντικείμενα του δέντρου μας. Το αντικείμενο Patient και το αντικείμενο History, ώστε να συνδυάσουμε πληροφορία από τα δύο αυτά elements. Η σελίδα Patient.html παρουσιάζεται ως εξής:



Όπως και στη προηγούμενη σελίδα έτσι και εδώ παράγουμε με ακριβώς τον ίδιο τρόπο μερικά στατιστικά πληροφοριακά στοιχεία για τους ασθενείς. InPatient είναι οι ασθενείς που αναρρώνουν εντός του νοσοκομείου, VisitingPatient είναι οι ασθενείς που θα έχουν μια μελλοντική επίσκεψη στο νοσοκομείο στο μέλλον και OutPatient είναι ένας πρώην ασθενής που έχουμε κρατήσει τα στοιχεία του. Και σε αυτή τη σελίδα χρησιμοποιήθηκαν δομές όπως <xsl:for-each>, <xsl:choose>, αρκετά hashtags που μας παραπέμπουν σε άλλες σελίδες. Λόγω του αρκετά σύνθετου κώδικα της XSLT δεν θα παρουσιαστούν παραδείγματα κώδικα της διότι η φύση του κώδικα για τη συγκεκριμένη σελίδα είναι ο ίδιος με τη Staff.html.

### Appointments.xslt - Appointments.html

Το αρχείο Appointments.xslt μετασχηματίζει το αρχικό μας αρχείο Hospital.xml στο αρχείο Appointments.html. Τα στοιχεία αντλούνται από το δέντρο Hospital.xml και παρουσιάζονται κατά φθίνουσα σειρά με βάση την ημερομηνία καταχώρησής τους. Δηλαδή από πιο πρόσφατο στο πιο παλιό. Αυτό επιτυγχάνεται με την xslt συνάρτηση <xsl:sort>. Παρακάτω βλέπουμε πως παρουσιάζεται το html αρχείο:



