



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**«Σχεδιασμός και πιλοτική ανάπτυξη
τουριστικών υπηρεσιών με βάση τα πρότυπα
του οργανισμού Hotel Technology next
generation»**

**ΜΕΤΑΠΤΥΧΙΑΚΗ
ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Γούργος Βασίλειος

**Επιβλέπων: Σγούρος Νικήτας-Μαρίνος
Καθηγητής Πανεπιστημίου Πειραιά**

ΠΕΙΡΑΙΑΣ 2009

Περιεχόμενα

Εισαγωγή	5
ΤΕΧΝΟΛΟΓΙΑ ΞΕΝΟΔΟΧΕΙΩΝ.....	6
Γενικά.....	6
Οφέλη του Converged Network	9
<i>Προσέλκυση και διατήρηση των φιλοξενούμενων μέσω των βελτιωμένων υπηρεσιών. ..</i>	<i>9</i>
<i>Δημιουργία και βελτίωση των επαναλαμβανόμενων πηγών εισοδήματος</i>	<i>10</i>
<i>Βελτίωση της παραγωγικότητας του προσωπικού</i>	<i>11</i>
<i>Ενιαία Διαχείριση υπηρεσιών για τη μείωση κόστους.....</i>	<i>12</i>
<i>Μείωση του κόστους.....</i>	<i>12</i>
<i>Τεχνολογική Εξέλιξη.....</i>	<i>13</i>
<i>The Case for Convergence.....</i>	<i>14</i>
Υπηρεσίες ψυχαγωγίας -δωματίων	15
Εισαγωγή στην HTML	16
XML (EXtensible Markup Language.).....	17
<i>Διαφορές μεταξύ XML και HTML</i>	<i>17</i>
<i>XML Elements</i>	<i>19</i>
<i>XML Attributes</i>	<i>19</i>
SOAP.....	20
<i>Εισαγωγή.....</i>	<i>20</i>
<i>Προγενέστερα πρωτόκολλα του SOAP ανταλλαγής μηνυμάτων.....</i>	<i>20</i>
<i>Μειονεκτήματα προηγούμενων τεχνολογιών</i>	<i>20</i>
<i>Κανόνες σύνταξης ενός SOAP μηνύματος.....</i>	<i>21</i>
<i>Γιατί να χρησιμοποιήσω το SOAP;</i>	<i>21</i>
<i>Δομή ενός μηνύματος SOAP</i>	<i>22</i>
<i>To SOAP Envelope</i>	<i>24</i>
<i>To SOAP Header</i>	<i>25</i>
<i>To SOAP Body.....</i>	<i>25</i>
<i>To SOAP Fault</i>	<i>25</i>
Javascript.....	26
<i>Τι είναι η Javascript;</i>	<i>26</i>
<i>Η Javascript είναι γλώσσα σεναρίου (script language).....</i>	<i>26</i>
<i>Ορισμός και εκτέλεση σεναρίων Javascript.....</i>	<i>26</i>

<i>Τι μπορεί να κάνει η Javascript;</i>	27
<i>Χαρακτηριστικά Javascript</i>	27
<i>Δηλώσεις μεταβλητών στη Javascript</i>	28
JSP (JavaServer Pages)	29
<i>Προκαθορισμένες μεταβλητές</i>	30
<i>Αρχιτεκτονική web εφαρμογής με JSP</i>	33
Servlets	35
<i>Τι είναι τα Java Servlets</i>	35
<i>Πλεονεκτήματα των servlets</i>	35
<i>Η μέθοδος init</i>	36
<i>Η μέθοδος destroy</i>	36
<i>Η λειτουργία ενός Servlet</i>	36
<i>Δομή ενός βασικού servlet</i>	37
<i>Μέθοδοι του αντικειμένου HttpSession</i>	39
<i>Ο κύκλος ζωής ενός Servlet</i>	40
Επιλογή Web Server	40
Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	42
<i>Γενικά</i>	42
<i>Booking</i>	42
<i>Customer</i>	44
<i>Employee</i>	46
<i>Room</i>	46
<i>tour_operator</i>	47
<i>usertable</i>	48
<i>package</i>	49
<i>taxrate</i>	49
<i>commission</i>	50
MySQL	51
<i>ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MySQL</i>	51
<i>Αναλυτική περιγραφή της εφαρμογής</i>	52
XML Μηνύματα της εφαρμογής	68
<i>SendingServlet.java</i>	68
<i>ReceivingServlet.java</i>	68

<i>SendingReservation.java</i>	70
<i>ReceivingReservation.java</i>	71
<i>SendingLogin.java</i>	71
<i>ReceivingLogin.java</i>	71
<i>SendingMonitorServlet.java</i>	72
<i>ReceivingMonitorServlet.java</i>	72
<i>TourOperSending.java</i>	73
<i>TourOperReceiving.java</i>	73
<i>SendingGroupServlet.java</i>	73
<i>ReceivingGroupServlet.java</i>	74
<i>SendingGroupReservation.java</i>	74
<i>ReceivingGroupReservation.java</i>	75
<i>SendingCheckinServlet.java</i>	75
<i>ReceivingCheckinServlet.java</i>	75
<i>SendCheckinServlet.java</i>	76
<i>ReceivCheckinServlet.java</i>	76
<i>SendingCheckoutServlet.java</i>	77
<i>ReceivingCheckoutServlet.java</i>	77
<i>SendCheckoutServlet.java</i>	78
<i>ReceivCheckoutServlet.java</i>	78
<i>SendingGroupCheckinServlet.java</i>	78
<i>ReceivingGroupCheckinServlet.java</i>	79
<i>SendGroupCheckinServlet.java</i>	79
<i>ReceivGroupCheckinServlet.java</i>	80
<i>SendingGroupCheckoutServlet.java</i>	80
<i>ReceivingGroupCheckoutServlet.java</i>	80
<i>SendGroupCheckoutServlet.java</i>	81
<i>ReceivGroupCheckoutServlet.java</i>	81
Επέκταση των δυνατοτήτων του προγράμματος	82
Βιβλιογραφία και διευθύνσεις στο Internet	83

Εισαγωγή

Η επιστήμη της Πληροφορικής έχει καταστεί μια από τις σημαντικότερες συνιστώσες των τουριστικών επιχειρήσεων, αφού ο σχεδιασμός και η υλοποίηση ενός ολοκληρωμένου πληροφοριακού συστήματος ψηφιακών τουριστικών υπηρεσιών αποτελεί ένα από τα βασικότερα στρατηγικά επιχειρηματικά εργαλεία γιατί η στρατηγική διοίκηση είναι εκείνη που οδηγεί στην λήψη των βέλτιστων στρατηγικών αποφάσεων και στην βραχυπρόθεσμη αλλά κυρίως μακροπρόθεσμη ανάπτυξη.

Στόχος είναι το παραδοσιακό μοντέλο διανομής τουριστικών προϊόντων και υπηρεσιών να μεταφερθεί αυτούσιο στο Internet, ώστε να είναι εφικτή η έγκαιρη ενημέρωση σχετικά με την ύπαρξη και τη διαθεσιμότητα τουριστικών προϊόντων και ο συντονισμός μεταξύ προσφοράς και ζήτησης. Ο έντονος ανταγωνισμός στο διεθνές τουριστικό επιχειρηματικό περιβάλλον επιβάλλει την άμεση χρήση σύγχρονων τεχνολογιών πληροφόρησης και επικοινωνίας.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας εφαρμογής για ξενοδοχειακές μονάδες η οποία θα μπορέσει να αξιοποιήσει καλύτερα τις δυνατότητες που δίνονται μέσω Διαδικτύου, χρησιμοποιώντας τις σύγχρονες τεχνολογίες με βάση τα πρότυπα για την Τεχνολογία Ξενοδοχείων - Επόμενη Γενιά (Hotel Technology - Next Generation - HTNG).

Η HTNG είναι μια μη κερδοσκοπικού χαρακτήρα οργάνωση η οποία δημιουργήθηκε το 2002 και έχει ως στόχο να δημιουργεί το κατάλληλο περιβάλλον συνεργασίας για την συνεχή ανάπτυξη πελατοκεντρικών τεχνολογιών νέας-γενιάς, για να ικανοποιήσει καλύτερα τις ανάγκες της παγκόσμιας ξενοδοχειακής κοινότητας (www.htng.org).

Η εφαρμογή (MyWebSite) που δημιουργήθηκε στα πλαίσια της Μεταπτυχιακής Διπλωματικής Εργασίας έχει τα εξής βασικά χαρακτηριστικά:

- Παρέχει λεπτομερή παρακολούθηση των κρατήσεων - πληρότητας - check in - Check out.
- Διαχειρίζεται τη διαθεσιμότητα δωματίων από εγγεγραμμένους χρήστες (ταξιδιωτικούς πράκτορες) οι οποίοι μπορούν να διενεργούν ομαδικές κρατήσεις.
- Πιστοποίηση δύο κατηγοριών χρηστών Tour Operators και υπαλλήλων του ξενοδοχείου μέσω username και password.

Η εφαρμογή χρησιμοποιεί τη Σχεσιακή Βάση Δεδομένων **hoteldb** για τη δημιουργία της οποίας έχει χρησιμοποιηθεί η MySQL.

ΤΕΧΝΟΛΟΓΙΑ ΞΕΝΟΔΟΧΕΙΩΝ

Γενικά

Τα δίκτυα υπολογιστών έχουν καταστεί το κεντρικό νευρικό σύστημα οποιασδήποτε σύγχρονης επιχείρησης. Για το λόγο αυτό υπάρχει η ανάγκη άμεσης προσαρμογής στα νέα δεδομένα της αγοράς και στις ανάγκες των οργανισμών που εξυπηρετούν. Αυτή η τεχνική εξέλιξης εστιάζεται στη σύγκλιση, δηλαδή τη συγχώνευση πολλών ανόμοιων δικτύων σε ένα. Η εξέλιξη της υποδομής των δικτύων στη βιομηχανία των ξενοδοχείων ακολουθεί αυτήν την πορεία.

Σήμερα, πολλές διαφορετικές δικτυακές αρχιτεκτονικές που χρησιμοποιούνται για φωνή, βίντεο και δεδομένα στα ξενοδοχεία, κινούνται σε ένα τέτοιο πλαίσιο ώστε να υπάρχει ένα ενιαίο δίκτυο με ένα ενιαίο πρωτόκολλο ικανό να υποστηρίζει πολλές εφαρμογές. Από τεχνικής άποψης, η σύγκλιση δικτύων είναι μια λογική εξέλιξη. Μπορούμε να πούμε ότι το converged network δίνει στους ξενοδόχους «περισσότερα ξοδεύοντας λιγότερα». Παρέχει περισσότερη ευελιξία, αξιοπιστία και ασφάλεια, ξοδεύοντας λιγότερα χρήματα.

Προς το τέλος της δεκαετίας του 90, πολλές επιχειρήσεις επενδύσανε έτσι ώστε να συγκλίνουν οι υποδομές των δικτύων τους. Αν και μερικές επιχειρήσεις αντιμετώπισαν δυσκολίες εξαιτίας των τεχνικών προκλήσεων, είδαν επιστροφή της επένδυσής τους. Η παραγωγικότητα τους αυξήθηκε εξαιτίας της μειωμένης πολυπλοκότητας. Σήμερα όπως προκύπτει από τα στοιχεία που υπάρχουν, η ιδέα των converged δικτύων έχει καταστεί κοινή πρακτική. Με την οικοδόμηση converged υποδομής δικτύων, τα ξενοδοχεία μπορούν να παρέχουν υψηλού επιπέδου και ασφαλείς υπηρεσίες, βελτιώνοντας το εισόδημα ανά διαθέσιμο δωμάτιο και την αποδοτικότητα του προσωπικού.

Το επίπεδο της τεχνικής υποδομής σε ολόκληρη τη βιομηχανία των ξενοδοχείων κυμαίνεται σε εξαιρετικά χαμηλά επίπεδα όταν συγκρίνεται με άλλες επιχειρήσεις. Η μεγάλη πλειοψηφία των ξενοδοχείων έχει μια σειρά από πληροφοριακά συστήματα, που κάθε ένα εκτελεί μια διαφορετική εργασία, είναι δαπανηρό και κατά ένα μεγάλο μέρος είναι ασυμβίβαστο το ένα με το άλλο. Πάρα πολύ συχνά, τα δίκτυα αυτά είναι διασκορπισμένα με αποτέλεσμα την έλλειψη υποστήριξης, την κακή διαχείριση και τις συχνές επιθέσεις ασφάλειας.

Τα προϊόντα τεχνολογίας μέσα στο ξενοδοχείο απαιτούν συχνά υποδομές που είναι ασύμβατες μεταξύ τους.

Τα converged πρότυπα υποδομής που αποτελούν τον κανόνα σε άλλες βιομηχανίες, συνδυάζουν κρίσιμες λειτουργίες όπως η επικύρωση χρηστών, η διαχείριση στοιχείων, η επεξεργασία κειμένου, το ηλεκτρονικό ταχυδρομείο και τη διαχείριση των εκτυπωτών μέσα από ένα ενιαίο σύστημα. Μερικά συστήματα απαιτούν τη δική τους ξεχωριστή χρήση τερματικών σταθμών και δικτύων, παρά το γεγονός ότι οι χρήστες μπορούν να έχουν PCs που συνδέονται με το τοπικό δίκτυο του ξενοδοχείου για να εκτελέσουν και άλλες λειτουργίες.

Τα σύγχρονα συστήματα τεχνολογίας απαιτούν μια συνεπή και αξιόπιστη υποδομή, που στηρίζεται σε τυποποιημένη τεχνολογία και μπορεί να χρησιμοποιηθεί από διάφορες εφαρμογές.

Τα βασικά στοιχεία της υποδομής των converged δικτύων είναι:

- Δίκτυα WANs και LANs
- Λειτουργικά συστήματα
- Διαχειριστική πλατφόρμα δεδομένων

Τα δευτερεύοντα στοιχεία της υποδομής των converged δικτύων περιλαμβάνουν τα γενικά επιχειρησιακά εργαλεία: το λογισμικό αυτοματισμού γραφείου, τα firewalls, τα αντιβιοτικά, και μια σειρά από βοηθητικές εφαρμογές.

Τα τελευταία χρόνια, έχει αυξηθεί η απαίτηση για την επέκταση των δικτύων μέσα στο δωμάτιο φιλοξενίας και σε δημόσια μέρη (περιοχές προηγουμένως εκτός ορίων δικτύων) βασισμένες σε σύγχρονες τεχνολογίες υπολογιστή. Σήμερα, αυτοί οι χώροι απαιτούν ένα πλήθος ενσύρματων και ασύρματων δικτύων παρέχοντας υπηρεσίες όπως η πρόσβαση στο Διαδίκτυο, η κινητή τηλεφωνική κάλυψη, συστήματα μηνύματος για απόκριση σε πραγματικό χρόνο, βίντεοπαραγγελία, ψηφιακή τηλεόραση, ψηφιακό CCTV, κλείδωμα πορτών, HVAC έλεγχος, ανίχνευση κατοχής δωματίου και πολλές άλλες απαραίτητες υπηρεσίες.

Πολλά παλαιότερα ξενοδοχεία, και μερικά από τα νεώτερα, δεν σχεδιάστηκαν έτσι ώστε η καλωδίωση να φθάνει στις αναγκαίες περιοχές, εμποδίζοντας την εύκολη επέκταση των απαραίτητων υπηρεσιών. Η προσαρμογή των υπαρχόντων τηλεφωνικών καλωδίων, των δικτύων καλωδιακής τηλεόρασης και των υψηλής τάσεως ηλεκτρικών καλωδίων απαιτούν σημαντικούς πόρους και μεγάλη προσπάθεια. Ασύρματες λύσεις που υιοθετούνται ευρέως μέσα σε οικιακά συγκροτήματα και

γραφεία, έχουν αποδειχθεί μη ικανοποιητικά στα ξενοδοχεία λόγω της πυκνότητας των δωματίων.

Μερικά από αυτά τα προβλήματα είναι σχετικά με την απουσία προτύπων υποδομής σε βιομηχανικό επίπεδο και άγνοιας του αντίκτυπου σε επίπεδο επιχείρησης ξενοδοχείων. Το ζητούμενο στην ξενοδοχειακή βιομηχανία είναι οι πληροφορίες να είναι άμεσα προσπελάσιμες ώστε να υποβοηθάτε συνεχώς το έργο της λήψης αποφάσεων από τη διοίκηση.

Η επένδυση στην υποδομή που απαιτείται για τα σημερινά συστήματα θα ήταν σημαντικά μικρότερη εάν είχε ως βάση της πρότυπα βιομηχανικού επιπέδου, που θα εξασφάλιζε τη συνεχή διαλειτουργικότητα. Επομένως είναι σκόπιμο να καθοριστεί μια κοινή πλατφόρμα που εγγυάται ένα κατώτατο επίπεδο διαλειτουργικότητας παρόλο που τα ξενοδοχεία είναι απρόθυμα να κάνουν τις απαραίτητες επενδύσεις.

Οφέλη του Converged Network

Στη σημερινή ανταγωνιστική τουριστική αγορά, τα ξενοδοχεία σε όλο τον κόσμο προσανατολίζονται στα ιδιαίτερα ευφυή converged networks προκειμένου να αυξήσουν την προσέλκυση περισσότερων πελατών. Ειδικότερα:

Προσέλκυση και διατήρηση των φιλοξενούμενων μέσω των βελτιωμένων υπηρεσιών.

Η προηγμένη επικοινωνία και οι ασύρματες ικανότητες που παρέχονται από ένα converged network επιτρέπει στα ξενοδοχεία να προσφέρουν νέες ή βελτιωμένες επικοινωνίες και υπηρεσίες ψυχαγωγίας. Ένα ενιαίο δίκτυο επιτρέπει τη διανομή των πληροφοριών των φιλοξενούμενων σε πολλές εφαρμογές ώστε να βελτιώσουν τις προσφερόμενες υπηρεσίες. Αυτό είναι πέρα από τις ικανότητες της σημερινής μη-συγκλιμένης τηλεφωνίας, Διαδικτύου και video. Υπάρχουν πολλά πράγματα που μπορούν να γίνουν μέσω ενός converged network όπως φαίνεται παρακάτω:

- **Περιβαλλοντικοί έλεγχοι:** Ρυθμίσεις για ελέγχους HVAC, ο φωτισμός δωματίου, χαρακτηριστικά άλλου δωματίου όπως το ανοίγμα/κλείσιμο κουρτινών μπορούν να ρυθμιστούν από το Front Desk. Αν οι προσωπικές προτιμήσεις του φιλοξενούμενου είναι γνωστές και διαθέσιμες σε μια γνωσιακή βάση, συνδεδεμένη με το PMS, η προσαρμογή του δωματίου μπορεί να είναι εξ ολοκλήρου αυτοματοποιημένη, κατά τη διάρκεια της διαδικασίας εισόδου.
- **Εξατομικευμένες υπηρεσίες τηλεφωνίας:** Με την αντικατάσταση των παραδοσιακών συστημάτων τηλεφωνίας με IP, **έτσι ώστε** οι ξενοδόχοι να μπορούν να προσφέρουν στους φιλοξενούμενους τους μια σειρά νέων υπηρεσιών οι οποίες είναι βασισμένες στην τηλεφωνία. Οι νέες υπηρεσίες, όπως τα μηνύματα, οι υπηρεσίες Διαδικτύου και η τηλεοπτική συνδιάσκεψη παρέχουν μεγάλες δυνατότητες στον φιλοξενούμενο. Η υψηλής ποιότητας speakerphones, οθόνες με δυνατότητες Web browsing και προσαρμοσμένες τηλεφωνικές επιλογές επιτρέπουν στο ξενοδοχείο να προσωποποιεί και να βελτιώνει τις υπηρεσίες σε σχέση με την τηλεφωνία.
- **Τηλεοπτική ψυχαγωγία επόμενης γενεάς:** Το ξενοδοχείο παρέχει τη δυνατότητα προηγμένων τηλεοπτικών υπηρεσιών μέσω ενός κοινού

δικτύου το οποίο μετασχηματίζει την τυποποιημένη τηλεόραση σε ένα προηγμένο κέντρο ψυχαγωγίας με τις προσαρμοσμένες προσφορές στο φιλοξενούμενο.

- **Δυνατότητα πρόσβασης στο Διαδίκτυο με υψηλή ταχύτητα** στο δωμάτιο αλλά και σε όλη την ξενοδοχειακή εγκατάσταση.

Δημιουργία και βελτίωση των επαναλαμβανόμενων πηγών εισοδήματος

- **Υπηρεσίες σύσκεψης:** Οι επιχειρησιακές διασκέψεις είναι μια σημαντική ευκαιρία παραγωγής εισοδήματος για πολλά ξενοδοχεία. Τα converged δίκτυα ξενοδοχείων παρέχουν στους επιχειρησιακούς ταξιδιώτες πρόσβαση Διαδικτύου υψηλής ταχύτητας σε όλο το ξενοδοχείο. Οι υπηρεσίες επικοινωνίας IP, όπως τα συστήματα τηλεφωνίας IP, παρέχουν τηλέφωνα με άριστες ικανότητες speakerphone, και υπηρεσίες σύσκεψης. Η Επικοινωνία των φιλοξενούμενων μέσω των συστημάτων του ξενοδοχείου επιτρέπουν στο ξενοδοχείο να ειδοποιούν τους φιλοξενούμενους για επερχόμενες συνόδους μέσω ενός τηλεφώνου ή μιας τηλεόρασης και παρέχουν την δυνατότητα για παρακολούθηση των τηλεοπτικών συνεδριάσεων μέσα στο δωμάτιο φιλοξενούμενων μέσω ζωντανών ή καταγραμμένων μεταδόσεων.
- **Η διαφήμιση του Ξενοδοχείου:** Τα ξενοδοχεία μπορούν να κερδίσουν νέες πηγές εισοδήματος με την παροχή guest-specific διαφήμιση ενημερώνοντας τους φιλοξενούμενους για τα καταστήματα και τις υπηρεσίες του ξενοδοχείου. Το converged δίκτυο επιτρέπει στα ξενοδοχεία να επεκτείνουν τη διαφήμιση μέσω ποικίλων μέσων: οθόνες πλάσματος στο λόμπι και τις περιοχές υποδοχής που επιδεικνύουν τις ανακοινώσεις ή τις διαφημίσεις των ξενοδοχείων, δικτυωμένα περίπτερα που βρίσκονται σε όλο το ξενοδοχείο, οθόνες τηλεφωνικής επίδειξης IP μέσα στα δωμάτια των φιλοξενούμενων και online portal υπηρεσιών.
- **Σύνδεση φωνής και δεδομένων για καταστήματα και εγκαταστάσεις:** Τα ξενοδοχεία μετατρέπονται σε παρόχους υπηρεσιών αξιοποιώντας την δυνατότητα που δίνεται από ένα converged network, παρέχοντας

τηλεφωνία, ασύρματη πρόσβαση στο Internet και γενικότερα όλες τις υπηρεσίες εκείνες οι οποίες μπορούν να αποτελέσουν πρόσθετες πηγές εισοδήματος.

Βελτίωση της παραγωγικότητας του προσωπικού

Το καλά εκπαιδευμένο προσωπικό είναι ένας από τους σημαντικούς παράγοντες που καθορίζουν το επίπεδο των παρεχόμενων υπηρεσιών. Ο φιλοξενούμενος θεωρεί αναπόσπαστο και παρά πολύ σημαντικό κομμάτι της φιλοξενίας το ίδιο το προσωπικό. Για την περαιτέρω βελτίωση της παραγωγικότητας του προσωπικού χρειάζεται:

- **Κινητικότητα εργατικού δυναμικού:** Το προσωπικό μπορεί να αυξήσει το επίπεδο της παρεχόμενης εξυπηρέτησης με το να είναι άμεσα διαθέσιμο οποιαδήποτε στιγμή του ζητηθεί από τον φιλοξενούμενο.
- **Προηγμένες τηλεφωνικές υπηρεσίες:** Με τα τηλέφωνα IP, τα ξενοδοχεία μπορούν να χρησιμοποιήσουν προηγμένες υπηρεσίες που παρέχουν ολοκλήρωση στα συστήματα επικοινωνιών έκτακτης ανάγκης και συστήματα διαχείρισης ιδιοκτησίας. Σε επείγουσα περίπτωση, το ξενοδοχείο μπορεί να μεταδώσει ζωντανό φωνητικό ή γραπτό μήνυμα άμεσα στα τηλέφωνα των φιλοξενούμενων.
- **Αυτόματη ασφάλεια δωματίων μέσω των βασικών συστημάτων καρτών:** Οι περισσότερες κορυφαίες επιχειρήσεις ξενοδοχείων έχουν εφαρμόσει βασικά συστήματα καρτών για να ελέγξουν την πρόσβαση στα δωμάτια φιλοξενούμενων. Το converged network βελτιώνει την ασφάλεια των φιλοξενούμενων, ενσωματώνοντας το σύστημα καρτών στο ολοκληρωμένο σύστημα ασφάλειας του ξενοδοχείου ελέγχοντας την πρόσβαση και τη θέση τους σε περιπτώσεις έκτακτης ανάγκης.
- **Τηλεοπτική σύσκεψη και δικτυακό περιεχόμενο για την κατάρτιση προσωπικού:** Ο μεγάλος αριθμός των υπαλλήλων και ο υψηλός κύκλος εργασιών του προσωπικού απαιτεί από τα ξενοδοχεία να αφιερώνουν πολύ χρόνο και χρήματα για την αποτελεσματική κατάρτιση του προσωπικού. Το τηλεοπτικό περιεχόμενο κατάρτισης μπορεί να αποθηκευτεί κεντρικά και παρέχεται στο προσωπικό μέσω ενός συστήματος βίντεο ή κατάρτισης τηλεδιάσκεψης ομάδων. Το τηλεοπτικό σύστημα διανομής για την κατάρτιση

προσωπικού επίσης παρέχει ζωντανές ή καταγραμμένες συνόδους διασκέψεων στο δωμάτιο φιλοξενουμένων.

Ενιαία Διαχείριση υπηρεσιών για τη μείωση κόστους

Οι υπηρεσίες που είναι διαθέσιμες μέσω των εφαρμογών τρέχουν μέσα από ένα converged network και χρησιμοποιούν το πρωτόκολλο IP όπως περιγράφονται παρακάτω:

- **IP Τηλεφωνία:** Ένα από τα πολλά οφέλη της τηλεφωνίας IP είναι ότι τα ξενοδοχεία δεν χρειάζονται ένα PBX (Private Branch Exchange - ιδιωτικό τηλεφωνικό κέντρο) ανά κτήριο. Μια επιχείρηση ξενοδοχείων αποτελούμενη από πολλές κτηριακές εγκαταστάσεις θα μπορεί να παρέχει τηλεφωνικές υπηρεσίες μέσω ενός κεντρικού υπολογιστή.
- **Απομακρυσμένη διαχείριση δικτύων:** Η κεντρική διαχείριση μας εξασφαλίζει μια σειρά από πλεονεκτήματα όπως: βελτίωση της ποιότητας των υπηρεσιών, καλύτερους χρόνους απόκρισης, αύξηση της παραγωγικότητας με αποτέλεσμα να την εξοικονόμηση χρόνου και χρημάτων.
- **Τηλεοπτική επιτήρηση IP:** Κάμερες επιτήρησης IP αντικαθιστούν το παραδοσιακό αναλογικό κλειστό τηλεοπτικό κύκλωμα (CCTV). Οι κασέτες καταγραφής βίντεο και η ανάγκη διαχείρισης και αποθήκευσής τους εξαλείφεται λόγω της αποδοτικότερης και πιο αποδοτικής διαχείρισης που μας παρέχει η νέα τεχνολογία.
- **Αυτοματοποίηση και έλεγχος:** Τα ξενοδοχεία μπορούν να βελτιώσουν το επίπεδο της συντήρησης και να μειώσουν λειτουργικά κόστη με τη χρησιμοποίηση των δικτυακών συστημάτων αυτοματοποίησης.
- **Κεντρικό σύστημα κρατήσεων:** Τα ξενοδοχεία μπορούν να κρατούν τα προσωπικά στοιχεία των φιλοξενουμένων σε ένα κεντρικό σύστημα κρατήσεων με αποτέλεσμα το προσωπικό να έχει πρόσβαση σε όλες τις διαθέσιμες πληροφορίες που αφορούν τον κάθε πελάτη παρέχοντας καλύτερες υπηρεσίες φιλοξενείας.

Μείωση του κόστους

Ένα converged δίκτυο εξαλείφει τα παραδοσιακά εμπόδια φωνής, βίντεο και δεδομένων. Οι δαπάνες του συνολικού κόστους του ξενοδοχείου μειώνονται

δεδομένου ότι αποφεύγεται η συντήρηση των πολυάριθμων υποδομών και απλουστεύεται η διαχείριση των συστημάτων.

- **Μείωση των δαπανών εξοπλισμού:** Η χρήση του converged δικτύου αποτρέπει τα ξενοδοχεία να επενδύουν σε συσκευές (όπως οι PBX) ή να διατηρούν τα χωριστά δίκτυα για το ISDN και το δωμάτιο ψυχαγωγίας. Αυτό μειώνει τη δαπάνη καλωδίωσης δεδομένου ότι δεν είναι απαραίτητο να εγκατασταθούν καλώδια διαφορετικών τύπων σε κάθε δωμάτιο, ώστε να υπάρχει η δυνατότητα τηλεφωνίας, ψυχαγωγίας και πρόσβασης στο Διαδίκτυο.
- **Μείωση των διαχειριστικών εξόδων δικτύων:** Ένα converged δίκτυο επιτρέπει στα ξενοδοχεία να μειώσουν τα διαχειριστικά έξοδα βελτιώνοντας την παραγωγικότητα του δικτύου. Δεν υπάρχει πλέον η ανάγκη συγκεκριμένων ομάδων υποστήριξης για συγκεκριμένα προϊόντα. Τα Ξενοδοχεία έχουν την δυνατότητα της απομακρυσμένης διαχείρισης.
- **Μείωση των δαπανών:** Η χρήση ενός δικτύου μεταφοράς δεδομένων που χρησιμοποιείται για τηλεφωνία μειώνουν το κόστος των κλήσεων. Αυτό προκαλεί επίσης μια σημαντική μείωση για διεθνείς κλήσεις.

Τεχνολογική Εξέλιξη

Σε ένα ξενοδοχείο υπάρχει η ανάγκη για πιο γρήγορη ανανέωση της τεχνολογικής υποδομής σε σχέση με την υλικοτεχνική. Ενώ ένα συγκρότημα γραφείων μπορεί να υποβληθεί σε μια τεχνική αναπροσαρμογή κάθε τρία με πέντε έτη, αυτό αδύνατο να επιτευχθεί σε ένα ξενοδοχείο. Η απαίτηση να διατηρηθεί ένας καθορισμένος αριθμός φιλοξενούμενων και το γεγονός της μείωσης της διαθεσιμότητας των δωματίων λόγω ανακαίνισης οδηγεί τη διοίκηση του ξενοδοχείου σε συνεχή αναβολή.

Ο προγραμματισμός του ξενοδοχείου μπορεί να καθυστερήσει την διαδικασία ανακαίνισης σε δεκαπέντε έτη. Μέσα στο διάστημα των δεκαπέντε ετών είναι πιθανό να χρειαστεί να καλύψουμε πέντε ή περισσότερες γενεές της τεχνολογίας. Τα ξενοδοχεία πρέπει να εγκαταστήσουν την αυριανή τεχνολογία σήμερα και πρέπει να είναι πολύ προσεκτικοί στις τεχνολογικές επιλογές τους.

The Case for Convergence

Είναι δυνατό να βρεθούν παλαιές υποδομές δικτύων που να μπορούν να ανταπεξέλθουν και σε σημερινές ανάγκες, αλλά αυτό είναι πάρα πολύ σπάνιο. Συνήθως, οι υποδομές είναι κατακερματισμένες, μη εύκολα διαχειρίσιμες και αναξιόπιστες, καλύπτοντας μόνο τις ανάγκες μιας ενιαίας εφαρμογής. Η έλλειψη ενός κοινού δικτύου, περιπλέκει τις δυνατότητες βελτίωσης, όπως η προσθήκη των κάμερας CCTV στα λόμπι των ανελκυστήρων ή προσθήκη πενήντα νέων καναλιών τηλεόρασης. Η έλλειψη των επενδύσεων σε υποδομή μπορεί να προέλθει από την έλλειψη αποτελεσματικού προγραμματισμού ή από μια γενική απροθυμία να ξοδευτούν χρήματα.

Οι περισσότεροι ξενοδόχοι που έχουν κάνει ήδη σημαντικές επενδύσεις, προσπαθούν να κάνουν χρήση της υπάρχουσας υποδομής κατά τη μετάβαση από τα χωριστά δίκτυα σε ένα converged πρότυπο δικτύωσης. Οι περισσότεροι προμηθευτές δικτύωσης έχουν δημιουργήσει προϊόντα για να διευκολύνουν την μετάβαση και να εξασφαλίσουν ότι ο νέος εξοπλισμός μπορεί να ενσωματωθεί εύκολα με την υπάρχουσα υποδομή. Η τεχνική υποδομή του ξενοδοχείου οδηγεί σε πρότυπα πολλών προμηθευτών που προσαρμόζονται γρήγορα στις εξελίξεις των επιχειρήσεων.

Υπηρεσίες ψυχαγωγίας -δωματίων

Τα συστήματα ψυχαγωγίας δωματίων παρέχουν στους φιλοξενούμενους πρόσβαση σε οπτικοακουστικές μορφές ψυχαγωγίας και πληροφορίες συμπεριλαμβανομένων των προϊόντων και των υπηρεσιών που περιγράφονται παρακάτω.

Free – To – Guest (FTG) – Το FTG δίνει στο φιλοξενούμενο τη δυνατότητα επιλογής από μια σειρά έτοιμων προγραμμάτων τηλεόρασης, χωρίς χρέωση.

Video on Demand (VOD) – VOD παρέχει στο φιλοξενούμενο την δυνατότητα πληρωμής περιεχομένου κατόπιν παράδοσης παραγγελίας. Το περιεχόμενο VOD παρέχει την πρόσβαση στο τηλεοπτικό περιεχόμενο από τις διάφορες πηγές συμπεριλαμβανομένων σημαντικά studios του Hollywood όπως το Paramount, Universal, MGM, Sony, Pixar, κ.λπ., αλλά και ανεξάρτητων παραγωγών, καθώς επίσης και οι δημιουργών όπως η HBO και η Showtime.

Promotional Video - Παραδίδεται δωρεάν στο θεατή και περιλαμβάνει την τηλεοπτική υποδοχή, διαφιμήσεις ξενοδοχείων, και προώθηση χώρων συναντήσεως όπως τα εστιατόρια, θέατρα, και άλλα θεάματα.

Music - Η μουσική μπορεί να περιλάβει τοπικούς και εθνικούς ραδιοσταθμούς, Διαδικτυακό ραδιόφωνο, δορυφορικό ραδιόφωνο, καθώς επίσης και μεγάλες βιβλιοθήκες αποθηκευμένης μουσικής και βίντεο μουσικής που παρουσιάζονται είτε για να προσελκύσουν στον θεατή του είτε έναντι αμοιβής.

Games - διάφοροι τύποι παιχνιδιών μπορούν να επιλεγθούν από ένα φιλοξενούμενο, συμπεριλαμβανομένων παιχνιδιών κονσόλας, διαδικτυακών παιχνιδιών κλπ.

Internet on TV – Παρέχει σε έναν φιλοξενούμενο πρόσβαση στο διαδίκτυο χρησιμοποιώντας την τηλεόραση ως συσκευή επίδειξης.

Guest Services - Οι υπηρεσίες φιλοξενούμενων είναι εφαρμογές που παρέχονται στο φιλοξενούμενο κατευθείαν από την τηλεόραση. Αυτό περιλαμβάνει τις εφαρμογές για κράτηση, υπηρεσία δωματίων κ.λπ.

Guest Device Connectivity - αντίθετα με τα σημερινά συστήματα ψυχαγωγίας ξενοδοχείων, τα μελλοντικά συστήματα δίνουν τη δυνατότητα στο φιλοξενούμενο να προσαρμόσει τις προσωπικές συσκευές του όπως media players, ψηφιακές φωτογραφικές μηχανές, προσωπικούς υπολογιστές, PD

Εισαγωγή στην HTML

Η **HTML** είναι το ακρωνύμιο των λέξεων **HyperText Markup Language** (γλώσσα μορφοποίηση υπερκειμένου) και είναι η βασική γλώσσα δόμηση σελίδων του *World Wide Web*. Δεν είναι μια γλώσσα προγραμματισμού, δεν μπορεί να παίρνει αποφάσεις, δεν μπορεί να κάνει πράξεις και αναθέσεις, δεν μπορεί να παρεκκλίνει από την ροή της, και άλλα πολλά πράγματα τα οποία δεν τα κάνει γιατί δεν είναι σχεδιασμένη για κάτι τέτοιο αλλά έχει κάποιους κανόνες οι οποίοι πρέπει να ακολουθηθούν προκειμένου να κατασκευαστεί ένα HTML document. Η HTML είναι μια περιγραφική γλώσσα που κάθε έκδοσή της ορίζετε από ένα Document Type Definition (DTD) του Standard Generalized Markup Language (SGML). Επιτρέπει την ενσωμάτωση ήχου και εικόνων στις web σελίδες. Αρχικά είχε κατασκευασθεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές. Η γλώσσα χρησιμοποιεί ένα αριθμό από **tags** για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (*links*) μετάβασης ανάμεσα των σελίδα, για την εισαγωγή εικόνων, ήχου κ.α. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (*tags*) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας.

Παράδειγμα Δομής ενός HTML κειμένου

```
<HTML>
<HEAD>
  <TITLE> Title </Title>
</HEAD>
<BODY>
  Body of HTML document
</BODY>
</HTML>
```


XML (EXtensible Markup Language.)

Η XML (eXtensible Markup Language) αναπτύχθηκε από το W3C's XML Working Group το 1996 και συνδυάζει την ισχύ και την επεκτασιμότητα της SGML (Standard Generalized Markup Language), από την οποία προέρχεται, με την απλότητα που απαιτεί η κοινότητα του Διαδικτύου.

- Η XML είναι συντομογραφία για το EXtensible Markup Language
- Markup: Είναι δηλωτική γλώσσα, χρησιμοποιεί ετικέτες με συγκεκριμένη σημασία.
- Extensible: Είναι επεκτάσιμη. Οι ετικέτες της XML δεν είναι προκαθορισμένες. Καθορίζονται από το χρήστη.
- Η XML σχεδιάστηκε για να περιγράφει τα δεδομένα.
- Η XML χρησιμοποιεί το **Document Type Definition (DTD)** ή το **XML Schema** για να περιγράψει τα δεδομένα
- Ένα XML κείμενο με ένα DTD ή ένα XML Schema σχεδιάστηκε για να **περιγράφει επαρκώς τον εαυτό του**
- Η XML χρησιμοποιείται για να αποθηκεύσουμε και να οργανώσουμε τα δεδομένα όχι για την παρουσίαση των δεδομένων, όπως κάνει η HTML.
- Η XML χρησιμοποιείται για ανταλλαγή δεδομένων
 - Το πρόγραμμά μου εξάγει τα δεδομένα του σε αρχείο XML.
 - Ένα άλλο πρόγραμμα μπορεί να διαβάσει τα XML αρχεία πιο εύκολα απ' ό,τι τα απλά αρχεία κειμένου.
 - Σε ένα XML έγγραφο, διατηρείται η δομή των δεδομένων μου.
- Η XML χρησιμοποιείται για αποθήκευση δεδομένων.
 - Τα XML αρχεία μπορούν να αντικαταστήσουν τις ΒΔ. Αρκεί να υπάρχουν τρόποι να κάνουμε επερωτήσεις και ενημερώσεις.

Διαφορές μεταξύ XML και HTML

- Η XML σχεδιάστηκε για τη μεταφορά δεδομένων.
- Η XML δεν αντικαθιστά την HTML αλλά τη συμπληρώνει.

- Η XML σχεδιάστηκε για να περιγράφει τα δεδομένα και εστιάζει στο τι είναι τα δεδομένα.
- Η HTML σχεδιάστηκε για να παρουσιάζει τα δεδομένα και εστιάζει στο πώς φαίνονται τα δεδομένα.
- Η HTML έχει να κάνει με την παρουσίαση των δεδομένων ενώ η XML έχει να κάνει με την περιγραφή των δεδομένων.

Δεν μπορούμε να αντικαταστήσουμε την XML με HTML. Μπορούμε, όμως, να συμπεριλάβουμε δεδομένα XML μέσα σε ετικέτες HTML και να τα προβάλλετε σε μια ιστοσελίδα. Η HTML περιορίζεται σε ένα προκαθορισμένο σετ ετικετών που μοιράζονται όλοι οι χρήστες ενώ η XML μάς επιτρέπει να δημιουργήσουμε οποιαδήποτε ετικέτα χρειαζόμαστε για να περιγράψετε τα δεδομένα μας και τη δομή τους.

Για παράδειγμα, υποθέστε ότι πρέπει να αποθηκεύσουμε και να μοιραστούμε πληροφορίες σχετικά με κατοικίδια ζώα.

Μπορούμε να δημιουργήσουμε τον παρακάτω κώδικα XML:

```
<?xml version="1.0"?>
<CAT>
  <NAME>Izzy</NAME>
  <BREED>Siamese</BREED>
  <AGE>6</AGE>
  <ALTERED>yes</ALTERED>
  <DECLAWED>no</DECLAWED>
  <LICENSE>Izz138bod</LICENSE>
  <OWNER>Colin Wilcox</OWNER>
</CAT>
```

Η πρώτη γραμμή δηλώνει την έκδοση της XML για το συγκεκριμένο έγγραφο και μπορεί να περιέχει πληροφορίες για την κωδικοποίηση (encoding) του εγγράφου.

Η δεύτερη γραμμή ορίζει το πρώτο στοιχείο του εγγράφου (τη ρίζα του) που είναι μοναδικό <CAT>.

Οι υπόλοιπες 7 γραμμές ορίζουν τα στοιχεία που βρίσκονται κάτω από τη ρίζα (NAME, BREED, AGE, ALTERED, DECLAWED, LICENSE, OWNER).

Η τελευταία γραμμή κλείνει το στοιχείο στη ρίζα. (</...>

XML Elements

- Όλα τα στοιχεία (elements) της XML πρέπει να έχουν ετικέτες ανοίγματος και κλεισίματος π.χ `<NAME>Izzy</NAME>`.
- Οι ετικέτες της XML είναι case sensitive (γίνεται διάκριση σε μικρά-κεφαλαία). Η ετικέτα `<Cat>` είναι διαφορετική από την `</cat>`
- Τα elements στην XML πρέπει να είναι σωστά φωλιασμένα.
`<CAT><NAME>Izzy</NAME></CAT>`.
- Όλα τα XML έγγραφα πρέπει να έχουν μία ρίζα. Η ρίζα πρέπει να είναι μοναδική. Όλα τα υπόλοιπα στοιχεία θα βρίσκονται φωλιασμένα σ' αυτήν.

XML Attributes

- Τα attributes χρησιμοποιούνται για να περιγράψουν τα χαρακτηριστικά των στοιχείων της XML.
- Περιέχονται πάντα στην αρχική ετικέτα του element σε ζευγάρια της μορφής: `όνομα="τιμή"`.
- Οι τιμές των χαρακτηριστικών πρέπει να βρίσκονται μέσα σε εισαγωγικά.
π.χ `<AGE value="20">`.
- Τα attributes έχουν περιορισμούς στον τύπο που πρέπει να έχουν τα δεδομένα που περιέχουν, για αυτό και προτιμούμε τα elements
- Τα attributes δεν μπορούν να περιγράψουν σύνθετες τιμές.

SOAP

Εισαγωγή

Το SOAP είναι ένα πρωτόκολλο βασισμένο στην XML το οποίο επιτρέπει στις εφαρμογές να ανταλλάσσουν πληροφορία πάνω από κοινώς χρησιμοποιούμενα πρωτόκολλα του διαδικτύου όπως είναι το HTTP.

Ειδικότερα το SOAP είναι:

- Το SOAP είναι συντομογραφία για το **Simple Object Access Protocol**
- Το SOAP είναι ένα **communication protocol**
- Το SOAP είναι **σχεδιασμένο για την επικοινωνία μεταξύ εφαρμογών**
- Το SOAP είναι **σχεδιασμένο για την αποστολή μηνυμάτων**
- Το SOAP είναι **σχεδιασμένο για την επικοινωνία μέσω του Internet**
- Το SOAP είναι **platform independent**
- Το SOAP είναι **language independent**
- Το SOAP βασίζεται **στην XML**
- Το SOAP είναι **απλό και επεκτάσιμο**
- Το SOAP επιτρέπει **την μεταφορά μηνυμάτων σε firewalls**
- Το SOAP θα αναπτυχθεί σαν ένα **W3C standard**

Προγενέστερα πρωτόκολλα του SOAP ανταλλαγής μηνυμάτων

- EDI
- CORBA
- COM και DCOM
- RMI
- XML-RPC → πρόδρομος του SOAP

Μειονεκτήματα προηγούμενων τεχνολογιών

- Κόστος υποδομής (EDI, CORBA)
- Κόστος και πολυπλοκότητα υλοποίησης (EDI, CORBA)
- Χρήση μόνο σε ιδιωτικά δίκτυα (EDI, CORBA, DCOM, RMI)
- Χρήση μόνο σε εφαρμογές που εκτελούνται στην ίδια πλατφόρμα (DCOM)

- Χρήση μόνο σε εφαρμογές γραμμένες στην ίδια γλώσσα προγραμματισμού (RMI)
- Μειωμένες λειτουργίες (XML-RPC)
- Έλλειψη ευελιξίας (EDI, CORBA, DCOM, RMI)

Κανόνες σύνταξης ενός SOAP μηνύματος

Παρακάτω αναφέρουμε μερικούς βασικούς κανόνες σύνταξης ενός μηνύματος SOAP:

- Ένα SOAP μήνυμα ΠΡΕΠΕΙ να γράφεται χρησιμοποιώντας XML
- Ένα SOAP μήνυμα ΠΡΕΠΕΙ να χρησιμοποιεί το SOAP Envelope namespace
- Ένα SOAP μήνυμα ΠΡΕΠΕΙ να χρησιμοποιεί το SOAP Encoding namespace
- Ένα SOAP μήνυμα ΔΕΝ πρέπει να χρησιμοποιεί αναφορά σε DTD
- Ένα SOAP μήνυμα ΔΕΝ πρέπει να περιέχει XML Processing Instructions

Γιατί να χρησιμοποιήσω το SOAP;

- Το SOAP είναι απλό. Άρα το κόστος και η πολυπλοκότητα υλοποίησης μειώνονται αισθητά.
- Το SOAP είναι ανεξάρτητο από πλατφόρμα και γλώσσα προγραμματισμού οπότε μπορεί να χρησιμοποιηθεί για επικοινωνία μεταξύ εφαρμογών γραμμένων για διαφορετικές πλατφόρμες και σε διαφορετικές γλώσσες προγραμματισμού.
- Το SOAP είναι ευέλικτο. Χρησιμοποιεί πρότυπα πρωτόκολλα όπως το HTTP και το SMTP ως μέσα μεταφοράς οπότε μπορεί να χρησιμοποιηθεί στο διαδίκτυο και να διαπερνά τύχη προστασίας χωρίς συμβιβασμούς στην ασφάλεια της υποδομής μιας επιχείρησης. Αυτό αυτομάτως μειώνει και σε ορισμένες περιπτώσεις εξαλείφει το κόστος υποδομής αφού οι περισσότερες επιχειρήσεις σήμερα έχουν και τον εξοπλισμό και την τεχνογνωσία για τη χρήση του διαδικτύου.
- Το SOAP είναι επεκτάσιμο. Αν και δεν προσφέρει τόσες πολλές λειτουργίες όσο άλλες τεχνολογίες όπως το CORBA και το DCOM επιτρέπει σε άλλα πρότυπα να το επεκτείνουν παρέχοντας υπηρεσίες που λείπουν από αυτό. Αυτό το χαρακτηριστικό αποδείχθηκε ίσως το σημαντικότερο γιατί επάνω του βασίζονται πολλές αναπτυσσόμενες τεχνολογίες των web services που προσφέρουν υπηρεσίες όπως «αξιοπιστία» (reliability), «δρομολόγηση» (routing) και «ασφάλεια» (security).

Δομή ενός μηνύματος SOAP

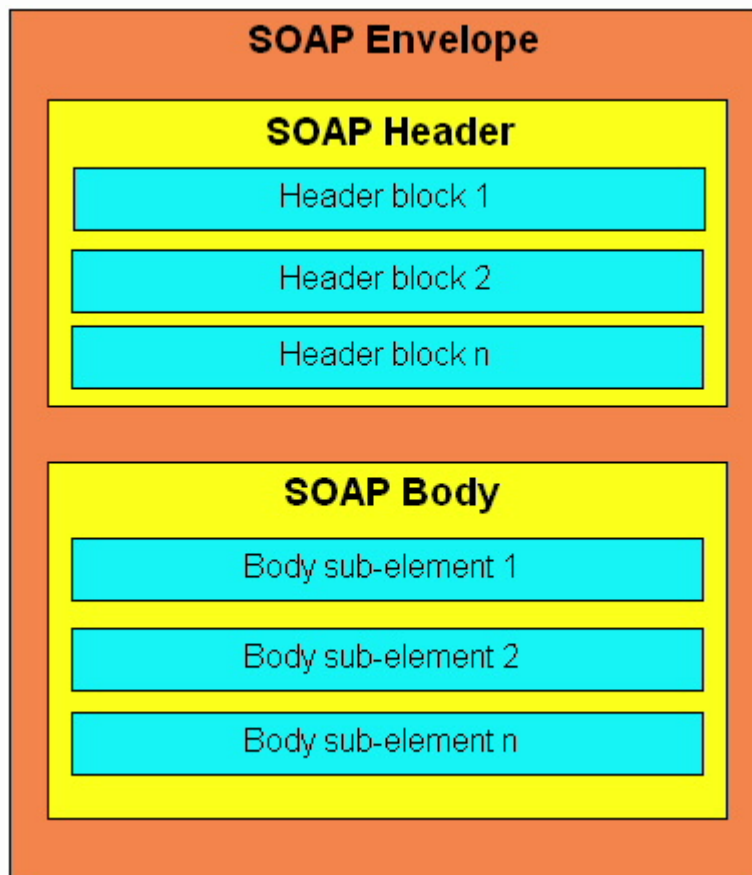
Ένα μήνυμα SOAP είναι ένα συνηθισμένο έγγραφο XML το οποίο περιέχει τα παρακάτω στοιχεία:

- Ένα Envelope element, το οποίο προσδιορίζει ότι το έγγραφο XML είναι ένα μήνυμα SOAP
- Ένα προαιρετικό Header (επικεφαλίδας) element, το οποίο περιέχει βοηθητικές πληροφορίες
- Ένα υποχρεωτικό Body element, το οποίο περιλαμβάνει την κύρια πληροφορία του μηνύματος.
- Ένα προαιρετικό Fault element, το οποίο περιέχει πληροφορίες για σφάλματα που προκλήθηκαν κατά την επεξεργασία του μηνύματος.

Η Δομή ενός μηνύματος SOAP φαίνεται παρακάτω:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
soap:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
...
...
</soap:Header>
<soap:Body>
...
...
</soap:Body>
</soap:Envelope>
```

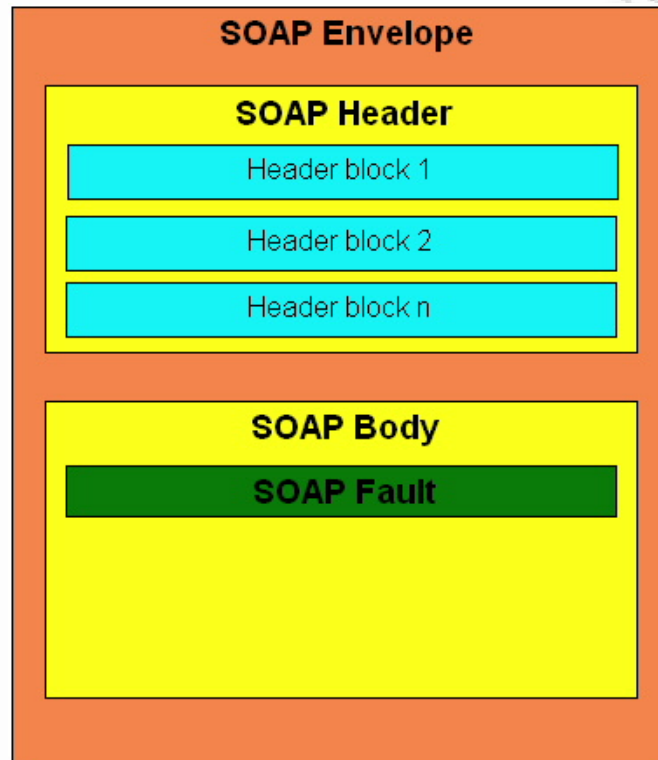
Σχηματικά θα μπορούσε να παρασταθεί ως εξής:



Όταν ένα μήνυμα SOAP περιέχει το στοιχείο Fault έχει την παρακάτω μορφή:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
soap:encodingStyle=" http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
...
...
</soap:Header>
<soap:Body>
  <soap:Fault>
    ...
    ...
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

Σχηματικά θα μπορούσε να παρασταθεί ως εξής:



To SOAP Envelope

Το element <Envelope> είναι το στοιχείο ρίζα ενός μηνύματος SOAP. Ορίζει το XML έγγραφο ως ένα μήνυμα SOAP.

Για το SOAP Envelope είναι απαραίτητα τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Envelope.
- Το όνομα του namespace του να είναι <http://www.w3.org/2003/05/soap-envelope>.
- Να περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Να περιέχει ένα ή δύο στοιχεία-παιδιά με την εξής σειρά :
 - Ένα προαιρετικό στοιχείο Header
 - Ένα απαραίτητο στοιχείο Body

To SOAP Header

Το προαιρετικό στοιχείο <Header> περιέχει επιπλέον, σχετικές με την εφαρμογή πληροφορίες που έχουν να κάνουν με το μήνυμα SOAP. Δεν είναι μέρος του προτύπου SOAP.

Για το SOAP Header είναι απαραίτητα τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Header.
- Το όνομα του namespace του να είναι "<http://www.w3.org/2003/05/soap-envelope>".
- Μπορεί να περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Μπορεί να περιέχει μηδέν ή περισσότερα στοιχεία-παιδιά ορισμένα με namespace.

To SOAP Body

Το υποχρεωτικό στοιχείο <Body> περιέχει το πραγματικό SOAP μήνυμα:

Για το SOAP Body είναι απαραίτητα τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Body.
- Το όνομα του namespace του να είναι "<http://www.w3.org/2003/05/soap-envelope>".
- Μπορεί να περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Μπορεί να περιέχει μηδέν ή περισσότερα στοιχεία-παιδιά ορισμένα με namespace.
- Μπορεί να περιέχει μηδέν ή περισσότερους κόμβους-παιδιά χαρακτήρων.

To SOAP Fault

Το στοιχείο <Body> μπορεί να περιέχει ένα <Fault> στοιχείο. Το <Fault> στοιχείο χρησιμοποιείται για να παρέχει πληροφορίες σχετικές με σφάλματα που συνέβησαν κατά την επεξεργασία ενός μηνύματος.

Για το SOAP Fault είναι απαραίτητα τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Fault.
- Το όνομα του namespace του να είναι "<http://www.w3.org/2003/05/soap-envelope>".
- Πρέπει να περιέχει δύο ή περισσότερα στοιχεία-παιδιά με την παρακάτω σειρά:
 - Ένα υποχρεωτικό στοιχείο Code.
 - Ένα υποχρεωτικό στοιχείο Reason.
 - Ένα προαιρετικό στοιχείο Node.
 - Ένα προαιρετικό στοιχείο Role.
 - Ένα προαιρετικό στοιχείο Detail.

Javascript

Τι είναι η Javascript;

Η JavaScript είναι μία διερμηνευμένη (interpreted) γλώσσα προγραμματισμού με ιδιότητες αντικειμενοστραφούς γλώσσας προγραμματισμού, χωρίς όμως να μπορεί να χαρακτηριστεί ως μία πλήρης αντικειμενοστραφής γλώσσα προγραμματισμού. Αναπτύχθηκε από τις Netscape και Sun Microsystems, διαθέσιμη από το 1995. Η Javascript είναι μια γλώσσα σεναρίου (script language) που χρησιμοποιείται για να δημιουργήσουμε διαδραστικές (interactive) ιστοσελίδες. Ο κώδικας JavaScript συνήθως ενσωματώνεται στον HTML κώδικα και εκτελείται από τον browser που «φορτώνει» το HTML έγγραφο. Η JavaScript υποστηρίζεται από όλους τους δημοφιλείς browsers, όπως ο Netscape και ο Internet Explorer, από την έκδοση 3.0 και έπειτα.

Η Javascript είναι γλώσσα σεναρίου (script language)

- Μια γλώσσα script επιτρέπει να γράψουμε ένα πρόγραμμα που δεν χρειάζεται να περάσει από τη διαδικασία της μεταγλώττισης (μετάφραση κώδικα σε γλώσσα μηχανής – εκτελέσιμο αρχείο), όπως συμβαίνει με τις κοινές γλώσσες προγραμματισμού (Pascal, Fortran, Visual Basic, ...).
- Ο κώδικας ενός script πρόγραμμα «διερμηνεύεται» άμεσα, από την εφαρμογή που το εκτελεί (π.χ. browser).
- Τυχόν συντακτική λάθη θα γίνουν αντιληπτά όταν «τρέξει» το script. Στην περίπτωση της Javascript, ο χειρισμός των σφαλμάτων γίνεται από τον browser που εκτελεί τον Javascript κώδικα

Ορισμός και εκτέλεση σεναρίων Javascript.

Κώδικας Javascript μπορεί να ενσωματωθεί σε ένα HTML έγγραφο με τους εξής τρόπους:

- Ο ορισμός ενός σεναρίου Javascript μέσα σε μια HTML ιστοσελίδα μπορεί να γίνει ανάμεσα σε ετικέτες: `<script language = "javascript"> </script>`.
- Τοποθέτηση του script ανάμεσα στα `<body>...</body>` tags. Τα σενάρια στο body εκτελούνται όταν φορτώνεται η σελίδα.
- Τοποθέτηση του script ανάμεσα στα `<head>...</head>` tags. Τα σενάρια στο head εκτελούνται με βάση τις ενέργειες του χρήστη.
- Αποθήκευση του script σε ένα **εξωτερικό αρχείο (*.js)**, καθώς και σύνδεση του γίνεται ως εξής:
`<script language = "javascript" src = "myexternal.js"> </script>`
- Συνδυασμός των προηγούμενων

Τι μπορεί να κάνει η Javascript;

- Η JavaScript παρέχει στους σχεδιαστές ιστοσελίδων ένα προγραμματιστικό εργαλείο
- Συνήθως, όσοι αναπτύσσουν HTML κώδικα δεν είναι προγραμματιστές, ωστόσο η JavaScript είναι μια scripting γλώσσα με πολύ απλή σύνταξη, άρα εύκολη στην εκμάθηση
- Η JavaScript μπορεί να εισάγει δυναμικό περιεχόμενο σε μια HTML σελίδα
- Για παράδειγμα, η παρακάτω JavaScript έκφραση: `document.write("<h1>" + name + "</h1>")` γράφει το περιεχόμενο της μεταβλητής name μέσα σε μια HTML σελίδα
- Ο JavaScript κώδικας μπορεί να αντιδράσει σε συμβάντα (events), δηλαδή να εκτελεστεί όταν κάτι συμβεί, π.χ. όταν:
 - Μια σελίδα φορτωθεί
 - Όταν ο χρήστης κάνει κλικ σε ένα πλήκτρο
 - Όταν ο χρήστης επιλέξει κάποιο στοιχείο μιας λίστας
- Ο JavaScript κώδικας μπορεί να διαβάσει αλλά και να μεταβάλει το περιεχόμενο ενός HTML στοιχείου (HTML elements), π.χ. να διαβάσει το κείμενο που έχει γράψει ο χρήστης σε ένα πεδίο κειμένου (textfield) και να κάνει κάτι με αυτό
- Εμφάνιση πληροφοριών / μηνυμάτων σε πλαίσια διαλόγου (dialog boxes), στη μπάρα κατάστασης ενός παραθύρου (status bar), κλπ.
- Εκτέλεση πολύπλοκων μαθηματικών υπολογισμών.

Χαρακτηριστικά Javascript

- Είναι ανεξάρτητη πλατφόρμας (όπως και η HTML)
- Εύκολη στη χρήση και την εκμάθηση
- Τα προγράμματα Javascript ενσωματώνονται σε HTML έγγραφα
- Τρέχει πολύ γρήγορα (δεν απαιτείται να σώσουμε κανένα ξεχωριστό αρχείο στο δίσκο μας)
- Είναι κατάλληλη για σχετικά απλές εφαρμογές
- Είναι βασισμένη στην έννοια του αντικειμένου (αντικειμενοστραφής γλώσσα)
- Χρησιμοποιείται κυρίως για προγραμματισμός στη μεριά του πελάτη (client), δηλαδή εκτελείται από το client application (browser) που «φορτώνει» το HTML έγγραφο.

Πως γράφουμε Javascript κώδικα;

```
<!-- Welcome.html-->
<html>
<body>
<script language="JavaScript">
document.write("Hello World with JavaScript");
</script>
</body>
</html>
```

Δηλώσεις μεταβλητών στη Javascript

- Στη Javascript χρησιμοποιούμε την έκφραση **var**
- Δεν δηλώνεται ο τύπος της μεταβλητής. Ο τύπος της μεταβλητής δηλώνεται έμμεσα, ανάλογα με την τιμή που της εκχωρούμε
- Τα ονόματα των μεταβλητών πρέπει να είναι περιγραφικά, δηλαδή να περιγράφουν αυτό που οι μεταβλητές αντιπροσωπεύουν
- Γίνεται διάκριση μεταξύ πεζών-κεφαλαίων στα ονόματα των μεταβλητών:
name ≠ Name

Παράδειγμα

```
<script language="JavaScript">
var name = "George";
var age = 22;
</script>
```

JSP (JavaServer Pages)

Οι *JSPs (JavaServer Pages)* είναι μια τεχνολογία που έχει δημιουργηθεί από την εταιρεία Sun Microsystems για να μπορεί να δημιουργεί ιστοσελίδες με δυναμικό περιεχόμενο (dynamic content). Οι JSP επιτρέπουν την εύκολη μίξη κώδικα HTML και κώδικα Java στην ίδια σελίδα. Οι JSP σελίδες γράφονται με έναν εκδότη κειμένου και τοποθετούνται συνήθως στον βασικό κατάλογο μιας web εφαρμογής (στο ίδιο σημείο που τοποθετούνται και οι HTML σελίδες). Όταν ζητάμε μία JSP σελίδα ο εξυπηρετητής διαδικτύου που χρησιμοποιούμε (π.χ. ο Tomcat) την μεταφράζει παρασκηνιακά σε servlet και την εκτελεί. Η μετάφραση σε servlet γίνεται μόνο την πρώτη φορά που ζητάμε την σελίδα και έτσι η επίδοση της εφαρμογής δεν επιβαρύνεται συγκριτικά με τα servlets. Τα διαφορετικά είδη του κώδικα Java που μπορούμε να έχουμε σε μία JSP σελίδα ονομάζονται στοιχεία σεναρίων (scripting elements).

Τα στοιχεία σεναρίων αυτά είναι τα εξής:

1. **Εκφράσεις (expressions):** Αυτές είναι εντολές Java που περιέχονται μεταξύ των σημάνσεων `<%=` και `%>`, αποτιμούνται άμεσα όπου εμφανίζονται και το αποτέλεσμά τους εισάγεται σαν κείμενο στο σημείου που εμφανίζονται.
2. **Τμήματα κώδικα (scriptlets):** Αυτά είναι τμήματα κώδικα Java που περιέχονται μεταξύ των σημάνσεων `<%` και `%>`. Οι δηλώσεις μεταβλητών, εντολές κλπ. που περιέχονται εκεί θα αποτελέσουν τμήμα της μεθόδου εξυπηρέτησης της αίτησης του πελάτη στο servlet που θα παραχθεί από την JSP σελίδα. Οι μεταβλητές που δηλώνονται στα τμήματα κώδικα, είναι τοπικές μεταβλητές της μεθόδου εξυπηρέτησης της αίτησης του πελάτη και άρα ασφαλής σε ταυτόχρονη προσπέλαση. Επίσης δημιουργούνται κάθε φορά που ζητάμε την σελίδα.
3. **Δηλώσεις (declarations):** Αυτές είναι δηλώσεις μεταβλητών αλλά και μεθόδων που περιέχονται μεταξύ των σημάνσεων `<%!` και `%>`. Οι δηλώσεις αυτών των μεταβλητών και μεθόδων έχουν εμβέλεια όλη την JSP σελίδα και όχι μόνο την μέθοδο εξυπηρέτησης του πελάτη. Αν πρόκειται για μεταβλητές επομένως δεν είναι ασφαλείς σε περίπτωση ταυτόχρονης προσπέλασης, αλλά επειδή ακριβώς δημιουργούνται μία φορά για κάθε αντικείμενο servlet είναι πιο αποδοτικό να χρησιμοποιούμε δηλώσεις όπου γίνεται. Βεβαίως λόγω

ταυτόχρονης προσπέλασης μερικές φορές είναι απαραίτητο να χρησιμοποιούμε συγχρονισμό για την προσπέλαση αυτών των μεταβλητών.

Οι JSPs μπορούν να περιλαμβάνουν τα εξής:

- στάνταρτ HTML tags (template text)
- στάνταρτ JSP tags (action elements)
- custom JSP tags
- scripting κώδικα

Προκαθορισμένες μεταβλητές

Υπάρχουν κάποιες προκαθορισμένες μεταβλητές οι οποίες δημιουργούνται αυτόματα στην μέθοδο εξυπηρέτησης μιας JSP σελίδας. Οι μεταβλητές αυτές είναι διαθέσιμες για τους προγραμματιστές JSP σελίδων και μπορούν να τις χρησιμοποιήσουμε μέσα από τμήματα κώδικα (scriptlets) και εκφράσεις (JSP expressions). Οι μεταβλητές ΔΕΝ μπορούν να χρησιμοποιηθούν σε δηλώσεις μεθόδων (declarations) μια και οι μέθοδοι που δηλώνονται δεν έχουν ορατότητα στις τοπικές μεταβλητές της μεθόδου εξυπηρέτησης. Αυτές οι προκαθορισμένες μεταβλητές είναι οι εξής:

1. **request:** Αυτή η μεταβλητή είναι το αντίστοιχο του HttpServletRequest αντικειμένου που είναι παράμετρος στην μέθοδο εξυπηρέτησης της αίτησης του χρήστη (π.χ. doGet ή doPost) για τα servlets. Αν, για παράδειγμα, θέλουμε να ανακτήσουμε την τιμή μιας παραμέτρου που δίνεται από μία φόρμα σε μία JSP σελίδα καλούμε την request.getParameter(<όνομα_παραμέτρου>).
2. **response:** Αυτή η μεταβλητή είναι το αντίστοιχο του HttpServletResponse αντικειμένου που είναι παράμετρος στην μέθοδο εξυπηρέτησης της αίτησης του χρήστη (π.χ. doGet ή doPost) για τα servlets.
3. **out:** Αυτή η μεταβλητή είναι το αντίστοιχο του PrintWriter που στα servlets ανακτούμε καλώντας την μέθοδο getWriter() στο αντικείμενο HttpServletResponse.

Η μεταβλητή out:

1) Χρησιμοποιείται σχεδόν αποκλειστικά μέσα σε τμήματα κώδικα (scriptlets) και αυτό διότι οι JSP εκφράσεις ούτως ή άλλως εξάγονται απευθείας στην HTML σελίδα που παράγεται.

- 2) Για την ακρίβεια η `out` είναι μία μεταβλητή τύπου `javax.servlet.jsp.JspWriter` μιας υποτάξης της τάξης `java.io.Writer`. Η τάξη `JspWriter` παρέχει την δυνατότητα προσωρινής αποθήκευσης (buffering). Αυτό σημαίνει πως όταν εξάγουμε κάτι από μία JSP σελίδα είτε γράφοντας απευθείας στο `out` είτε από μία JSP έκφραση αυτό αποθηκεύεται στην προσωρινή περιοχή αποθήκευσης (buffer) και γράφεται όταν γεμίσει η προσωρινή περιοχή αποθήκευσης.
4. **session:** Αυτό είναι ένα προ-δημιουργημένο αντικείμενο τύπου `HttpSession`. Παρέχει τις γνωστές μεθόδους `getAttribute` και `setAttribute` για την ανάκτηση και αποθήκευση ιδιοτήτων αντίστοιχα στο `session` του χρήστη με τον δικτυακό τόπο. Αν θέλετε να απαγορεύσετε την συμμετοχή μιας JSP σελίδας σε `sessions` μπορείτε να χρησιμοποιήσετε την οδηγία σελίδας `page` θέτοντας την ιδιότητα `session` στην τιμή `false` ως εξής `<%@page session="false"%>`. Σημειώστε πως η εξ ορισμού τιμή είναι `true`. Το νόημα της οδηγίας αυτής είναι πως αν το `session` είναι `true` τότε η σελίδα έχει πρόσβαση στο αντίστοιχο `HttpSession` αντικείμενο αν υπάρχει και αν δεν υπάρχει δημιουργείται. Αν το `session` είναι `false` τότε αν στην σελίδα υπάρχουν κλήσεις μεθόδων στο `session` αντικείμενο η μετάφραση της JSP σελίδας σε `servlet` θα αποτύχει (compile-time error).
5. **application:** Αυτή η μεταβλητή είναι το `ServletContext` αντικείμενο που στα `servlets` επιστρέφεται από την μέθοδο `ServletConfig.getContext()`. Το αντικείμενο αυτό έχει πολλές χρήσιμες μεθόδους όπως οι ακόλουθες:
- `getAttribute` και `setAttribute`: οι μέθοδοι αυτές έχουν την ίδια σημασία με τις αντίστοιχες του `session` μόνο που αφορούν ιδιότητες που είναι προσπελάσιμες από όλα τα `servlets` και `JSPs` μιας εφαρμογής και δεν έχουν σχέση με το `session` ενός χρήστη.
 - `getInitParameter`: η μέθοδος αυτή ανακτά παραμέτρους αρχικοποίησης που έχουν τεθεί για όλη την εφαρμογή στο `web.xml` της εφαρμογής με το στοιχείο `<context-param>`. Οι διαφορά των παραμέτρων αρχικοποίησης μιας εφαρμογής από τις παραμέτρους αρχικοποίησης ενός `servlet` είναι πως οι παράμετροι αρχικοποίησης της εφαρμογής είναι προσπελάσιμες από όλα τα `servlets` και `JSPs` μιας εφαρμογής. Ένα παράδειγμα μιας παραμέτρου εφαρμογής στο `web.xml` της είναι το εξής:

```
<context-param>
    <param-name>registryMachine</param-name>
    <param-value>//localhost</param-value>
</context-param>
```

Για να ανακτήσει μια JSP την τιμή της παραμέτρου θα καλούσε την μέθοδο `application.getInitParameter("registryMachine")`.

- `getRequestDispatcher`: επιστρέφει ένα `RequestDispatcher` αντικείμενο για το `path` που δίνεται ως παράμετρος. Η JSP σελίδα μπορεί μετά να καλέσει τις μεθόδους του `RequestDispatcher` για προώθηση στο στοιχείο αυτό (μέθοδος `forward()`) ή συμπερίληψη του αποτελέσματος στο αποτέλεσμα της σελίδας (μέθοδος `include()`).

- `log()`: εισάγει το μήνυμα που δίνεται ως παράμετρος στο αρχείο συμβάντων της εφαρμογής (log file). Στον tomcat τα αρχεία συμβάντων αποθηκεύονται στον φάκελο `%TOMCAT_HOME%\logs`.

6. **config**: Αυτό είναι το αντικείμενο `ServletConfig` της σελίδας. Η κύρια χρήση του είναι η ανάκτηση παραμέτρων αρχικοποίησης για την σελίδα.
7. **pageContext**: Αυτό το αντικείμενο είναι ένα αντικείμενο της τάξης `javax.servlet.jsp.PageContext`. Η κυριότερη δυνατότητα που παρέχεται για τους προγραμματιστές JSP σελίδων είναι η αποθήκευση και η ανάκτηση αντικειμένων μέσω των μεθόδων `getAttribute` και `setAttribute` αλλά σε τέσσερις διαφορετικές εμβέλεις (scopes). Οι εμβέλεις μπορεί να είναι `page`, `application`, `session` και `request`. Σε κάθε περίπτωση χρησιμοποιούνται οι αντίστοιχοι μηχανισμοί των `servlets`.
8. **page**: Αυτό το αντικείμενο είναι συνώνυμο με το `this` για το παραγόμενο `servlet`. Στην αρχή της `_jspService` του παραγόμενου `servlet` υπάρχει η δήλωση "`Object page = this;`". Βεβαίως μπορείτε να χρησιμοποιήσετε και απευθείας το `this`.

Οι JSPs έχουν τα παρακάτω πλεονεκτήματα:

- Είναι μέρος του κώδικα που συνοδεύει την πλατφόρμα της Java.
- Είναι ανεξάρτητες της αρχιτεκτονικής του συστήματος στο οποίο εκτελούνται.

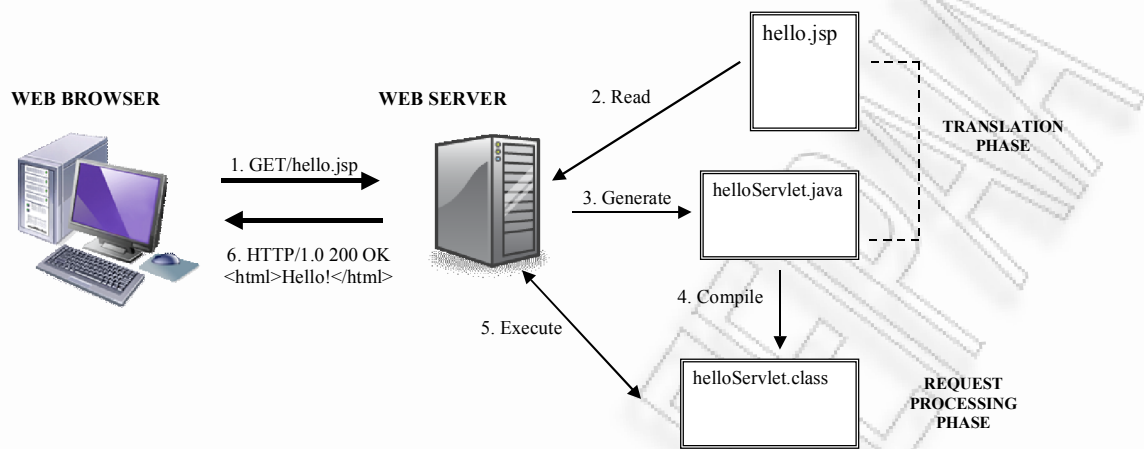
- Μπορούν να «επικοινωνήσουν» πολύ εύκολα με άλλες κλάσεις ή οντότητες της Java.
- Μπορούν να εξυπηρετούν ταυτόχρονα πολλαπλές αιτήσεις διαφορετικών χρηστών και επιπλέον έχουν τη δυνατότητα να τις συγχρονίζουν.
- Έχουν τη δυνατότητα να διαβιβάζουν τις αιτήσεις των χρηστών σε άλλους servers ή JSPs.
- Παρέχουν όλες τις δυνατότητες των φαινομενικά ισχυρότερων servlets, αφού αυτόματα μεταφράζονται σε servlets που αναλαμβάνουν να φέρουν σε πέρας την απαίτηση.
- Δίνουν τη δυνατότητα στον προγραμματιστή να δημιουργήσει πολύ εύκολα τόσο στατικές όσο και δυναμικές οντότητες, με μια πιο φυσική προσέγγιση από άλλες τεχνολογίες.
- Εκτός από έξοδο σε HTML μορφή, τα JSPs μπορούν να έχουν και έξοδο σε XML, ή ακόμα και σε WML μορφή, η οποία είναι απαραίτητη για την επικοινωνία με τους χρήστες μέσω του WAP.
- Συνοδεύεται από πλήρη και κατατοπιστική τεκμηρίωση (documentation), η οποία παρέχεται δωρεάν μέσω του World Wide Web.
- Υποστηρίζονται από τη μεγάλη πλειονότητα των servers που κυκλοφορούν στην αγορά λογισμικού.

Αρχιτεκτονική web εφαρμογής με JSP

Τρία συστατικά στοιχεία

- Servlet - είναι υπεύθυνο για την επεξεργασία των requests από τον browser (κέντρο ελέγχου)
- JSP - είναι υπεύθυνη για την παρουσίαση του αποτελέσματος (presentation)
- JavaBean - είναι υπεύθυνο για την υλοποίηση της business λογικής της εφαρμογής

Φάσεις επεξεργασίας μιας JSP σελίδας



Οι JSPs είναι μια εφαρμογή στην πλευρά του server (server-side application), που σημαίνει ότι δέχονται μια αίτηση (request) και παράγουν μια απόκριση ή απάντηση (response). Σε γενικές γραμμές, οι αιτήσεις γίνονται από έναν Web client και η απόκριση είναι ένα παραγόμενο HTML έγγραφο (ιστοσελίδα) το οποίο στέλνεται πίσω στον Web client. Επειδή οι JSPs είναι μια εφαρμογή στην πλευρά του server, έχουν πρόσβαση σε πηγές (resources) στον server, όπως είναι τα Servlets, JavaBeans, EJBs, αλλά και σε βάσεις δεδομένων.

Όταν γράφουμε ένα JSP, είναι ευκολότερο να γράψουμε πρώτα τον HTML κώδικα και μετά να εισάγουμε τον κώδικα της Java για να δημιουργήσουμε το δυναμικό περιεχόμενο. Ακολουθεί ένα απλό παράδειγμα JSP το οποίο δείχνει πώς μπορεί να εισαχθεί κώδικας της Java σ' ένα JSP. Αυτό που κάνει αυτό το JSP είναι ότι εισάγει την τρέχουσα ημερομηνία και ώρα του συστήματος του Web server σε μια HTML σελίδα.

```
<html>
<head><title> Program Title </title></head>
<body>
    <center>
        Τρέχουσα Ημερομηνία και Ώρα : <%= new java.util.Date() %>
    </center>
</body>
</html>
```

Servlets

Τι είναι τα Java Servlets

Τα Java Servlets είναι μία web τεχνολογία βασισμένη στη Java με την οποία μπορούμε να αναπτύξουμε server-side εφαρμογές για το Internet. Ένα servlet είναι μία μικρή τάξη της Java που περιέχεται σε ένα container. Το container διαχειρίζεται το servlet. Το container μπορεί να είναι ταυτόχρονα και web server ή να είναι ένα add-on component για ένα web server χωρίς δυνατότητες εκτέλεσης servlet. Για παράδειγμα ο Tomcat που θα συζητήσουμε στη συνέχεια, είναι container και ταυτόχρονα web server. Παρόλα αυτά, υπάρχει η δυνατότητα να χρησιμοποιηθεί ο Tomcat μαζί με τον Apache ή τον IIS οι οποίοι δεν μπορούν να εκτελέσουν servlets.

Πλεονεκτήματα των servlets

Με τα servlets μπορούμε να αναπτύξουμε πολύ ισχυρές web εφαρμογές που έχουν τα εξής πλεονεκτήματα:

- **Αποδοτικότητα & Αντοχή** Είναι πολύ εύκολο να προσθέσουμε state σε εφαρμογές διαδικτύου. Το HTTP είναι ένα stateless πρωτόκολλο. Αυτό σημαίνει πως σε κάθε get που κάνει ο χρήστης ενός browser για μία ιστοσελίδα, για τον web server είναι σαν να ξεκινά από την αρχή. Με τα servlets υπάρχει η έννοια της συνεδρίας (session) με την οποία μπορούμε να κρατήσουμε πληροφορίες που σχετίζονται με την συνεδρία του χρήστη με το σύστημα που μπορούν να διατηρούνται ανάμεσα από διαδοχικά get. Αυτό είναι απαραίτητο σε πολλές εφαρμογές. Κλασικό παράδειγμα είναι το shopping cart σε μία εφαρμογή πώλησης προϊόντων μέσω Internet, στο οποίο ο χρήστης τοποθετεί προϊόντα καθώς πηγαίνει από σελίδα σε σελίδα. Όπως είναι αυτονόητο, το shopping cart θα πρέπει να συγκρατεί τα περιεχόμενά του ανάμεσα από διαδοχικές αιτήσεις για νέες σελίδες.
- **Εκμεταλλεύονται πλήρως** το Java API, rmi, database connectivity.
- Είναι **platform independent** - “Write once Serve Everywhere” - Portability Έχουμε ανεξαρτησία από Λειτουργικό Σύστημα, μια και τα Servlets εκτελούνται από την Java Virtual Machine.
- **Ασφάλεια Κώδικα** (Garbage Collector, Exception Handling)
- **Κομψότητα** (Elegance) (Object Oriented, Clean Code, Modular, Simple)
- **Εύκολο integration** νέων υπηρεσιών.

Η μέθοδος init

Η μέθοδος `init` μπορεί να χρησιμοποιηθεί για την αρχικοποίηση ενός `Servlet`. Εκτελείται πριν εκτελεσθεί η `doGet` ή η `doPost` ή οποιαδήποτε άλλη μέθοδος, την πρώτη φορά που δημιουργείται ένα στιγμιότυπο. Υπάρχουν δύο παραλλαγές αυτής της μεθόδου που είναι οι ακόλουθες:

1. `public void init()`,
2. `public void init(ServletConfig)`

Η πρώτη μορφή χρησιμοποιείται όταν δεν χρειάζεται στην μέθοδο `init` να αναφερθούμε στο αντικείμενο `ServletConfig` το οποίο μπορεί να χρησιμοποιηθεί για πρόσβαση σε πληροφορίες σχετικά με τις ρυθμίσεις ενός `Servlet`. Η δεύτερη έχει πρόσβαση στο αντικείμενο αυτό μια και περνιέται ως παράμετρος, αλλά θα πρέπει **πάντα να θυμηθούμε σαν πρώτη εντολή να καλέσουμε την `super.init(ServletConfig)`**. Και οι δύο μορφές καλούνται από το `Servlet container` ακριβώς μετά την δημιουργία του πρώτου στιγμιότυπου και πριν εξυπηρετηθεί η πρώτη αίτηση.

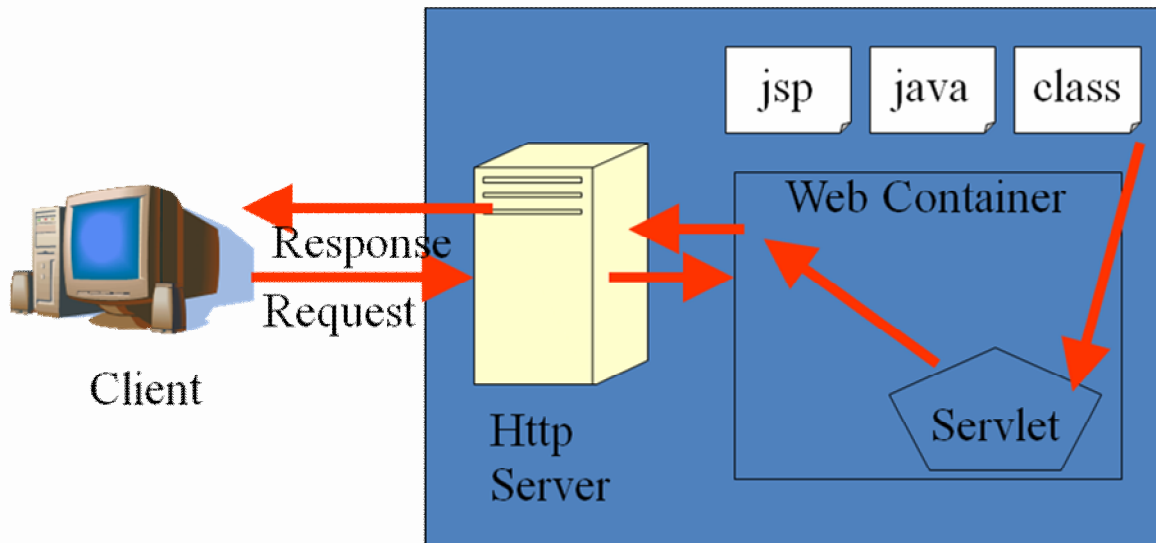
Η μέθοδος destroy

Η μέθοδος `destroy` καλείται από το `Servlet container` πριν ένα `Servlet` γίνει μη διαθέσιμο και αφού έχουν τερματισθεί όλα τα `threads` που το προσπελάζουν ή έχει λήξει η περίοδος μη χρήσης του στιγμιότυπου για μια εφαρμογή. Η μέθοδος αποτελεί το κατάλληλο σημείο για εργασίες εκκαθάρισης (`clean up`), όπως αποδέσμευση μνήμης ή χειριστών αρχείων (`file handles`), καθώς και για την αποθήκευση τυχόν στοιχείων που θα πρέπει να αποθηκευθούν και βρίσκονται στην μνήμη του `Servlet`.

Η μέθοδος **`destroy`** κάνει τα εξής:

- Αυτή η μέθοδος αποτελεί το **κατάλληλο σημείο για εργασίες εκκαθάρισης** (`clean up`), όπως:
 1. **αποδέσμευση** μνήμης,
 2. **αποθήκευση** τυχόν στοιχείων που θα πρέπει να αποθηκευθούν και βρίσκονται στην μνήμη του `Servlet`.

Η λειτουργία ενός Servlet



Δομή ενός βασικού servlet

Θα εξετάσουμε την βασική δομή ενός servlet όπως είναι το HelloWorld servlet :

```
public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World</title>");
        out.println("</head>");
        out.println("<body bgcolor='white'>");
        out.println("<h1>Hello World</h1>");
        out.println("</body>");
        out.println("</html>"); } }
```

Παρατηρείστε πως το HelloWorld επεκτείνει το HttpServlet. Η τάξη HttpServlet του πακέτου javax.servlet.http είναι υποτάξη της τάξης javax.servlet.GenericServlet και είναι μία αφηρημένη τάξη που θα πρέπει να επεκτείνουμε για να δημιουργήσουμε ένα HttpServlet κατάλληλο για χρήση σε ένα website. Οι τάξεις που επεκτείνουν το HttpServlet θα πρέπει να υπερβούν τουλάχιστον μία από τις μεθόδους:

- doGet
- doPost
- doPut
- doDelete,
- init,
- destroy και
- getServletInfo.

Η χρήση των doPut και doDelete είναι σπάνια. Οι μέθοδοι που υπερβαίνουμε είναι μία από τις doPost ή doGet και ίσως να χρησιμοποιήσουμε και τις init, destroy και getServletInfo. Η μέθοδος doGet χρησιμοποιείται όταν τα δεδομένα μίας φόρμας αποστέλλονται με προσάρτηση στο URL (method="get"). Η μέθοδος doPost χρησιμοποιείται όταν τα δεδομένα μίας φόρμας αποστέλλονται σε ένα ανεξάρτητο ρεύμα (method="post"). Εδώ επεκτείνουμε την μέθοδο doGet διότι δεν αποστέλλονται δεδομένα στο servlet. Η doGet (και doPost) παίρνουν ως παραμέτρους δύο αντικείμενα, το request και το response, που υλοποιούν τις δύο διασυνδέσεις javax.servlet.http.HttpServletRequest και javax.servlet.http.HttpServletResponse αντίστοιχα. Τα δύο αυτά αντικείμενα παρέχουν μεθόδους για την επεξεργασία της αίτησης του χρήστη (HttpServletRequest) αλλά και την αποστολή της απόκρισης του servlet (HttpServletResponse).

Βασικά οι μέθοδοι που καλούμε σε αυτό το Servlet είναι τρεις:

- Η μέθοδος setContentType του αντικειμένου response θα πρέπει να κληθεί για να προσδιορισθεί ο τύπος της απόκρισης. Στην περίπτωση μας η απόκριση θα είναι "text/html". Αυτήν η μέθοδος θέτει το ContentType response Header στην HTML σελίδα που σχηματίζεται και θα δει ο χρήστης. Ο πιο κοινός τύπος απόκρισης είναι "text/html" αλλά δεν είναι ο μόνος. Για παράδειγμα αν η απόκριση ήταν μια εικόνα ο τύπος θα έπρεπε να είναι "image/gif" και αν ήταν ένα έγγραφο του excel "application/vnd.ms-excel". Κάτι χρήσιμο για την εμφάνιση Ελληνικών είναι πως με την setContentType μπορούμε να προσδιορίσουμε και το charset δηλαδή την κωδικοσελίδα στην οποία ο PrintWriter θα μετατρέψει τους Unicode χαρακτήρες της java για να τους αποστείλλει στον χρήστη. Για παράδειγμα θα μπορούσαμε να προσδιορίσουμε το ContentType ως "text/html; charset=UTF-8" για να σταλλεί το αποτέλεσμα σε μορφή Unicode και "text/html; charset=ISO-8859-7" για να

σταλλεί το αποτέλεσμα σε κωδικοποίηση Ελληνική ISO. Σε περίπτωση που δεν προσδιορίσουμε το charset τότε χρησιμοποιείται το ISO Latin1.

- Η μέθοδος `getWriter` όπως ήδη αναφέραμε, επιστρέφει ένα `PrintWriter` αντικείμενο, στο παράδειγμά μας το `out`, το οποίο είναι ένα ρεύμα χαρακτήρων εισόδου του αντικειμένου `response`, στο οποίο γράφουμε (με την `println`) την HTML σελίδα. Ο `PrintWriter` μετατρέπει τους Unicode χαρακτήρες της Java στο charset που προσδιορίσαμε με την `setContentType` και τους εξάγει στον browser.
- Τέλος καλούμε την `println` του αντικειμένου `out` για να σχηματίσουμε την ιστοσελίδα-αποτέλεσμα που θα δει ο χρήστης. Στις συμβολοσειρές της `out.println` συνήθως περιέχονται και ετικέτες HTML για την μορφοποίηση του κειμένου της ιστοσελίδας που θα δει ο χρήστης.

Ανάκτηση της συνόδου του χρήστη

Για να βρούμε μέσα από ένα `servlet` ποιο είναι το `HttpSession` αντικείμενο που σχετίζεται με την αίτηση ενός χρήστη, χρησιμοποιούμε την μέθοδο `HttpServletRequest.getSession`.

- `HttpSession getSession()`
- `HttpSession getSession(boolean create)`

Μέθοδοι του αντικειμένου `HttpSession`

- Η `setAttribute` εισάγει ένα αντικείμενο (το *value*) στην σύνοδο του χρήστη, χρησιμοποιώντας το όνομα *name* γι' αυτό το αντικείμενο.
 - Αν υπήρχε ένα αντικείμενο με αυτό το όνομα ήδη, το αντικείμενο αυτό αντικαθίσταται από το νέο αντικείμενο.
- `public java.util.Enumeration getAttributeNames()`
 - Επιστρέφει μία `Enumeration` με `Strings` που είναι τα ονόματα όλων των αντικειμένων που έχουν εισαχθεί σε μία σύνοδο.
- `public void setMaxInactiveInterval(int interval)`
 - Αυτή η μέθοδος θέτει το μέγιστο διάστημα μεταξύ δύο διαδοχικών αιτήσεων για το `servlet` για το οποίο το `servlet container` δεν ακυρώνει την σύνοδο. (TTL)
- `public void removeAttribute(String name)`

- Αφαιρεί το αντικείμενο το οποίο είχε εισαχθεί στην σύνοδο με το όνομα που δίνεται ως παράμετρος.
- Αν η σύνοδος δεν έχει αντικείμενο που έχει συσχετισθεί με αυτό το όνομα τότε η μέθοδος αυτή δεν κάνει τίποτα.
- **public String getId()**
 - Επιστρέφει ένα String με το αναγνωριστικό της συνόδου (session ID). Αυτό παράγεται όταν δημιουργείται η σύνοδος.
- **public void invalidate()**
 - Αυτή η μέθοδος ακυρώνει την σύνοδο και αποδεσμεύει τα ονόματα και τα αντικείμενα που είχαν εισαχθεί σε αυτήν.
- **public boolean isNew()**
 - Αυτή η μέθοδος επιστρέφει true εφόσον μία σύνοδος που επιστράφηκε από την `HttpServletResponse.getSession()` είναι νέα και false διαφορετικά.

Ο κύκλος ζωής ενός Servlet

Κάθε servlet έχει τον ίδιο κύκλο ζωής:

- Ο server το φορτώνει (load) και το αρχικοποιεί (initialize).
- Το servlet δέχεται μηδέν ή και περισσότερα client requests
- Ο server το κάνει remove (ορισμένοι servers εκτελούν αυτό το βήμα μόνο όταν κάνουν shut down)

Επιλογή Web Server

Ως Web Server χρησιμοποιήθηκε ο Tomcat 6.0.18. Τα πλεονεκτήματά του σε σχέση με άλλους Web Server που κυκλοφορούν στην αγορά λογισμικού είναι τα ακόλουθα:

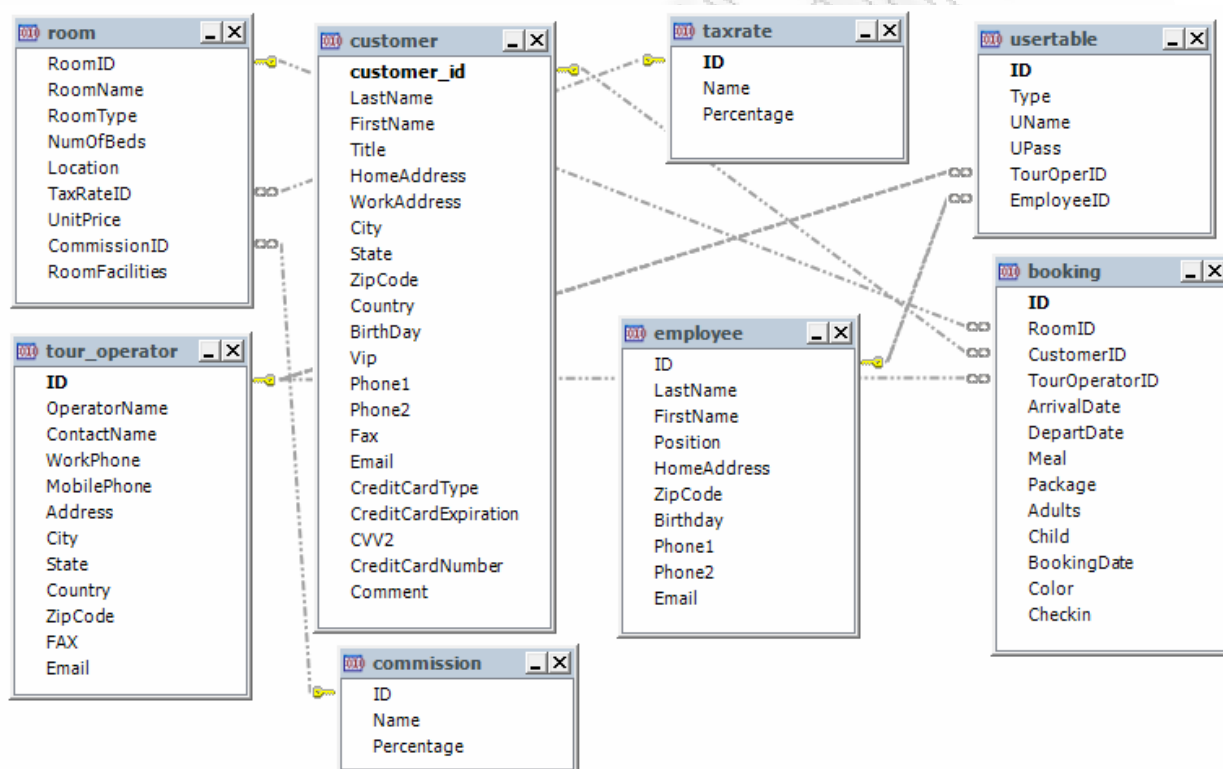
- Παρέχει υποστήριξη για εφαρμογές που κάνουν χρήση της τεχνολογίας των Java Server Pages (JSP).
- Αποτελεί έναν αρκετά σταθερό Web Server.

- Συνοδεύεται από πλήρη και κατατοπιστική τεκμηρίωση (documentation), η οποία παρέχεται δωρεάν μέσω του World Wide Web.
- Είναι ένας από τους πιο διαδεδομένους Web Servers που κυκλοφορούν στην αγορά λογισμικού.
- Διανέμεται δωρεάν μέσω του World Wide Web.
- Ο παραγόμενος κώδικας βέβαια είναι πολύ εύκολο να χρησιμοποιηθεί και σε άλλους Web Servers, αρκεί βέβαια υποστηρίζουν την τεχνολογία των Java Server Pages.

Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Γενικά

Για τη δημιουργία της βάση δεδομένων χρησιμοποιήθηκε MySql. Το όνομα της βάσης δεδομένων είναι «hoteldb» η οποία περιέχει πίνακες δεδομένων. Μέσα σε κάθε πίνακα δηλώνω τον τύπο κάθε πεδίου (π.χ. Char, int, date) και το πρωτεύον του κλειδί (primary key), το οποίο μπορεί να αποτελείται από ένα πεδίο με την προϋπόθεση να είναι μοναδικό (unique).



Περιγραφή του κάθε πίνακα και των πεδίων τους ξεχωριστά:

Booking

Στον πίνακα αυτό καταχωρούμε στοιχεία που αφορούν κρατήσεις μεμονωμένου χρήστη αλλά και ταξιδιωτικού πράκτορα καθώς επίσης και στοιχεία που αφορούν το Check in καθώς επίσης και πρόσθετες πληροφορίες που έχουν σχέση με το είδος της διατροφής που επιθυμεί ο πελάτης, εάν έχει επιλέξει κάποιο συγκεκριμένο πακέτο προσφοράς και τέλος εάν υπάρχει κάποιο παιδί το οποίο θα φιλοξενηθεί στο δωμάτιο ώστε να ληφθεί υπόψη από τη διεύθυνση του ξενοδοχείου για να συμπεριληφθεί σε αυτό ένα επιπλέον κρεβάτι.

Field	Type	Key	Default	Extra
ID	int	PRI		auto_increment
RoomID	int		0	
CustomerID	int		0	
TourOperatorID	int		0	
ArrivalDate	date			
DepartDate	date			
Meal	varchar(2)			
Package	varchar(5)		null	
Adults	varchar(2)			
Child	varchar(3)		No	
BookingDate	date			
Color	varchar(7)			
Checkin	date			

Αναλυτική περιγραφή των πεδίων του πίνακα Booking:

Πεδίο	Περιγραφή
ID	Κωδικός κράτησης δωματίου
RoomID	Κωδικός δωματίου
CustomerID	Κωδικός πελάτη
TourOperatorID	Κωδικός τουριστικού πράκτορα
ArrivalDate	Ημερομηνία Αφίξης
DepartDate	Ημερομηνία Αναχώρησης
Meal	Είδος Διατροφής
Package	Κωδικός πακέτου
Adults	Αριθμός ενηλίκων
Child	Ύπαρξη παιδιού (Ναι/Όχι)
BookingDate	Ημερομηνία κράτησης
Color	Χρώμα με το οποίο δηλώνουμε σε ποια κατάσταση βρίσκεται το δωμάτιο κάποια συγκεκριμένη ημερομηνία.
Checkin	Ημερομηνία Checkin

Customer

Ο πίνακας αυτός αφορά την καταχώρηση πελατών εισάγοντας τα προσωπικά τους στοιχεία.

Field	Type	Key	Default	Extra
customer_id	int	PRI		auto_increment
LastName	varchar(20)			
FirstName	varchar(20)			
Title	varchar(20)			
HomeAddress	varchar(20)			
WorkAddress	varchar(20)			
City	varchar(20)			
State	varchar(20)			
ZipCode	varchar(10)			
Country	varchar(20)			
BirthDay	date			
Vip	varchar(10)			
Phone1	varchar(15)			
Phone2	varchar(15)			
Fax	varchar(15)			
Email	varchar(30)			
CreditCardType	varchar(20)			
CreditCardExpiration	varchar(20)			
CVV2	varchar(20)			
CreditCardNumber	varchar(20)			
Comment	varchar(10)			

Αναλυτική περιγραφή των πεδίων του πίνακα customer:

Πεδίο	Περιγραφή
customer_id	Κωδικός πελάτη
LastName	Επώνυμο
FirstName	Όνομα
Title	Τίτλος
HomeAddress	Διεύθυνση Κατοικίας
WorkAddress	Διεύθυνση Εργασίας
City	Πόλη
State	
ZipCode	Ταχυδρομικός κώδικας
Country	Χώρα
BirthDay	Ημερομηνία Γέννησης
Vip	Εάν είναι Vip(Ναι/όχι)
Phone1	Αριθμός Τηλεφώνου
Phone2	Αριθμός Τηλεφώνου
Fax	Αριθμός Fax
Email	Ηλεκτρονικό ταχυδρομείο
CreditCardType	Ο τύπος της πιστωτικής Κάρτας
CreditCardExpiration	Expiration Κάρτας
CVV2	Και το CVV2
CreditCardNumber	Αριθμός της κάρτας
Comment	Σχόλια

Employee

Ο πίνακας αυτός αφορά την καταχώρηση πελατών εισάγοντας τα προσωπικά τους στοιχεία.

Field	Type	Key	Default	Extra
ID	int	PRI		auto_increment
LastName	varchar(20)			
FirstName	varchar(20)			
Position	varchar(20)			
HomeAddress	varchar(20)			
ZipCode	varchar(10)			
BirthDay	date			
Phone1	varchar(15)			
Phone2	varchar(15)			
Email	varchar(30)			

Αναλυτική περιγραφή των πεδίων του πίνακα customer:

Πεδίο	Περιγραφή
ID	Κωδικός πελάτη
LastName	Επώνυμο
FirstName	Όνομα
Position	Θέση
HomeAddress	Διεύθυνση Κατοικίας
ZipCode	Ταχυδρομικός κώδικας
BirthDay	Ημερομηνία Γέννησης
Phone1	Αριθμός Τηλεφώνου
Phone2	Αριθμός Τηλεφώνου
Email	Ηλεκτρονικό ταχυδρομείο

Room

Ο πίνακας αυτός περιλαμβάνει όλη την πληροφορία που έχει σχέση με την δυναμικότητα του ξενοδοχείου. Παρέχει πληροφορία για τον τύπο του δωματίου των αριθμό των κρεβατιών τον όροφο που βρίσκεται το κάθε δωμάτιο, την τιμή του ανά ημέρα, καθώς και πληροφορίες για τους ταξιδιωτικούς πράκτορες για το ποσοστό της προμήθειας για κάθε ένα ξεχωριστά.

Field	Type	Key	Default	Extra
RoomID	int	PRI	0	
RoomName	varchar(5)			
RoomType	varchar(4)			
NumOfBeds	int			
Location	varchar(10)			
TaxRateID	int			
UnitPrice	double			
CommissionID	int			
RoomFacilities	varchar(100)			

Αναλυτική περιγραφή των πεδίων του πίνακα Room:

Πεδίο	Περιγραφή
RoomID	Κωδικός δωματίου
RoomName	Όνομα δωματίου
RoomType	Τύπος δωματίου
NumOfBeds	Αριθμός Κρεβατιών
Location	Όροφος δωματίου
TaxRateID	Κωδικός Φόρου
UnitPrice	Τιμή δωματίου ανά ημέρα
CommissionID	Κωδικός προμήθειας
RoomFacilities	

tour_operator

Ο πίνακας αυτός αφορά την καταχώρηση των προσωπικών στοιχείων των τουριστικών πρακτόρων.

Field	Type	Key	Default	Extra
ID	int	PRI		auto_increment
OperatorName	varchar(20)			
ContactName	varchar(20)			
WorkPhone	varchar(15)			
MobilePhone	varchar(15)			

Address	varchar(20)			
City	varchar(20)			
State	varchar(20)			
Country	varchar(15)			
ZipCode	varchar(10)			
FAX	varchar(15)			
Email	varchar(30)			

Αναλυτική περιγραφή των πεδίων του πίνακα tour_operator:

Πεδίο	Περιγραφή
ID	Κωδικός τουριστικού πράκτορα
OperatorName	Όνομα τουριστικού πράκτορα
ContactName	Υπεύθυνος επικοινωνίας
WorkPhone	Αριθμός τηλεφώνου εργασίας
MobilePhone	Αριθμός κινητού τηλεφώνου
Address	Διεύθυνση
City	Πόλη
State	
Country	Χώρα
ZipCode	Ταχυδρομικός κώδικας
FAX	Αριθμός FAX
Email	Ηλεκτρονικό ταχυδρομείο

usertable

Ο usertable είναι ένας πίνακας με τη βοήθεια του οποίου ορίζονται οι εξουσιοδοτήσεις που έχει να κάνει με τον ρόλο και τον τύπο του χρήστη

Field	Type	Key	Default	Extra
ID	int	PRI		auto_increment
Type	varchar(3)			
UName	varchar(10)			
UPass	varchar(10)			
TourOperID	int			
EmployeeID	int			

Αναλυτική περιγραφή των πεδίων του πίνακα usertable:

Πεδίο	Περιγραφή
ID	Κωδικός χρήστη
Type	Τύπος του χρήστη
UName	Το username του χρήστη
UPass	Το password του χρήστη
TourOperID	Κωδικός του τουριστικού πράκτορα
EmployeeID	Κωδικός του υπαλλήλου

package

Στον πίνακα αυτό ορίζονται όλα τα διαθέσιμα πακέτα του ξενοδοχείου

Field	Type	Key	Default	Extra
ID	varchar(5)	PRI		
Name	varchar(20)			
Price	float			
PackageDays	varchar(2)			

Αναλυτική περιγραφή των πεδίων του πίνακα package:

Πεδίο	Περιγραφή
ID	Κωδικός πακέτου
Name	Όνομα πακέτου
Price	Τιμή πακέτου
PackageDays	Ημέρες του πακέτου

taxrate

Στον πίνακα αυτό ορίζονται όλα τα ποσοστά φορολόγησης του ξενοδοχείου

Field	Type	Key	Default	Extra
ID	int	PRI		
Name	varchar(20)			
Percentage	float		0	

Αναλυτική περιγραφή των πεδίων του πίνακα taxrate:

Πεδίο	Περιγραφή
ID	Κωδικός φόρου
Name	Όνομα φόρου
Percentage	Ποσοστό φόρου

commission

Στον πίνακα αυτό ορίζονται όλα τα ποσοστά προμήθειας του κάθε δωματίου ανάλογα τον τύπο του.

Field	Type	Key	Default	Extra
ID	int	PRI		
Name	varchar(20)			
Percentage	float			

Αναλυτική περιγραφή των πεδίων του πίνακα commission:

Πεδίο	Περιγραφή
ID	Κωδικός προμήθειας
Name	Όνομα προμήθειας
Percentage	Ποσοστό προμήθειας

MySQL

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων μας επιτρέπει να αποθηκεύουμε, να αναζητούμε, να ταξινομούμε και να ανακαλούμε τα δεδομένα μας αποτελεσματικά. Ο διακομιστής MySQL ελέγχει την πρόσβαση στα δεδομένα μας, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα και να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Χρησιμοποιεί την **SQL (Structured Query Language)** την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL, Microsoft SQL και Oracle.

ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MySQL

Απόδοση

Η MySQL είναι γρήγορη, πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

Χαμηλό κόστος

Η MySQL είναι διαθέσιμη δωρεάν, με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος, εάν προμηθευτούμε εμπορική άδεια, αν απαιτηθεί από την εφαρμογή μας.

Ευκολία Χρήσης

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν έχετε χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων δεν θα έχετε πρόβλημα να προσαρμοστείτε σε αυτό.

Μεταφερσιμότητα

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα όπως Unix, Microsoft Windows.

Κώδικας Προέλευσης

Μπορούμε να τροποποιήσουμε τον κώδικα προέλευσης της MySQL.

Αξιοπιστία

Η έκδοση MySQL 5 είναι ικανή να υποστηρίξει πολύ μεγάλα projects με υψηλή αξιοπιστία.

Αναλυτική περιγραφή της εφαρμογής

Στην παρακάτω εικόνα βλέπουμε την αρχική σελίδα του MyWebSite

HTNG HOTEL

Home Page
Reservation
PhotoGallery
Members

Hotel Name: HTNG Hotel
Hotel Address: Karaiskou 112 street 18535 Athens
Local Area: Center
City: Athens
Accommodation: Breakfast
Check-In: 13:00 pm
Check-Out: 11:00 am
Rooms: 100

16 km
1 km

Hotel Facilities

General
Restaurant – Bar – 24-Hour Front Desk – Newspapers – Garden – Terrace – Non-Smoking Rooms – Elevator – Express Check-In/Check-Out – Safety Deposit Box – Soundproofed Rooms – Heating – Luggage Storage – Gay Friendly

Services
Room Service – Meeting/Denquet Facilities – Business Centre – Laundry – Dry Cleaning – Breakfast in the Room – Ironing Service – Currency Exchange – Shoe Shine – Packed Lunches – Car Rental – Tour Desk – Fax/Photocopying

Internet
Wireless internet is available in public areas and is free of charge.
Wired internet is available in the hotel rooms and is free of charge.

Guest Parking
No parking available.

Hotel Policies

Check In
11:00 hours
Check Out
07:00 – 12:00 hours
Cancellation
If cancelled up to 1 day before the date of arrival, no fee will be charged.
If cancelled later or in case of no-show, the first night will be charged.

Children and extra beds
All children under 2 years stay free of charge when using existing bedding.
All children under 2 years stay free of charge for cots.
There is no capacity for extra beds in the room.
Maximum capacity of babycots in a room is 1.
Extra beds and baby cots are upon request and need to be confirmed by the hotel.

Pets
Pets are not allowed.

Accepted credit cards

- American Express
- Visa
- Euro/Mastercard
- Diners Club

The hotel reserves the right to pre-authorise credit cards prior to arrival.

Food & beverage

A drink to lighten up and pleasures for the palette. A garden to refresh, pure joy for the eyes. A cozy bar and a restaurant with fragrances and flavors from Greece and the world, to suit all tastes. It is here where will find the ideal place for relaxing with friends or meeting business associates in a warm, informal atmosphere.

Abundant light is the dominative element of the restaurant "Acropolis" where the soft colors are relaxing and rejuvenating. A place to meet and enjoy the dishes the chef creates especially for you in an absolutely genuine Greek environment in the center of the old Athens.

Your coffee and your drink become a real escape at the "Parthenon" bar which is situated in the atrium of the hotel vested by beautiful plants.

General Comment
The hotel is placed at the foothill of the Acropolis next to the Ancient city of Athens (Plaka) which is full of tavernas, cafes and clubs. You are in the heart of Athens and everything is so close to visit.

Travel Information
Parthenon Hotel is about 50 minutes away by taxi from the Eleftherios Venizelos Airport and the fair costs approx. 20 euros.
We would not recommend you to use a car since the location of the hotel (Acropolis) has been transformed to pedestrian precinct and it will be very difficult to find a parking space (parkings are very expensive in the area) but if you insist, take the Attiki odos motorway towards Elefsina and then take the exit No 11 (Kifissias) towards Athens. Then follow the Kifissias Avenue which turns to Vassilisis Sofias Avenue and then turns in to Vassileos Konstadinou. Turn right at Vassilikos Oligas Avenue and then left to Vassilisis Amalias Avenue. The hotel is located on your right hand at the Dionisiou Aeropagitou pedestrian precinct which leads to the Acropolis.
Alternatively, take the Subway from the Airport towards the terminal Monastiraki. Get off at Syntagma station. Then take the red line towards Agios Dimitrios and get off at the Acropolis Station. The hotel is a 150 meters walk.

goubil@yahoo.gr

Ολοκληρώθηκε

Internet | Προστατευμένη κατάσταση λειτουργίας: Ανενεργή

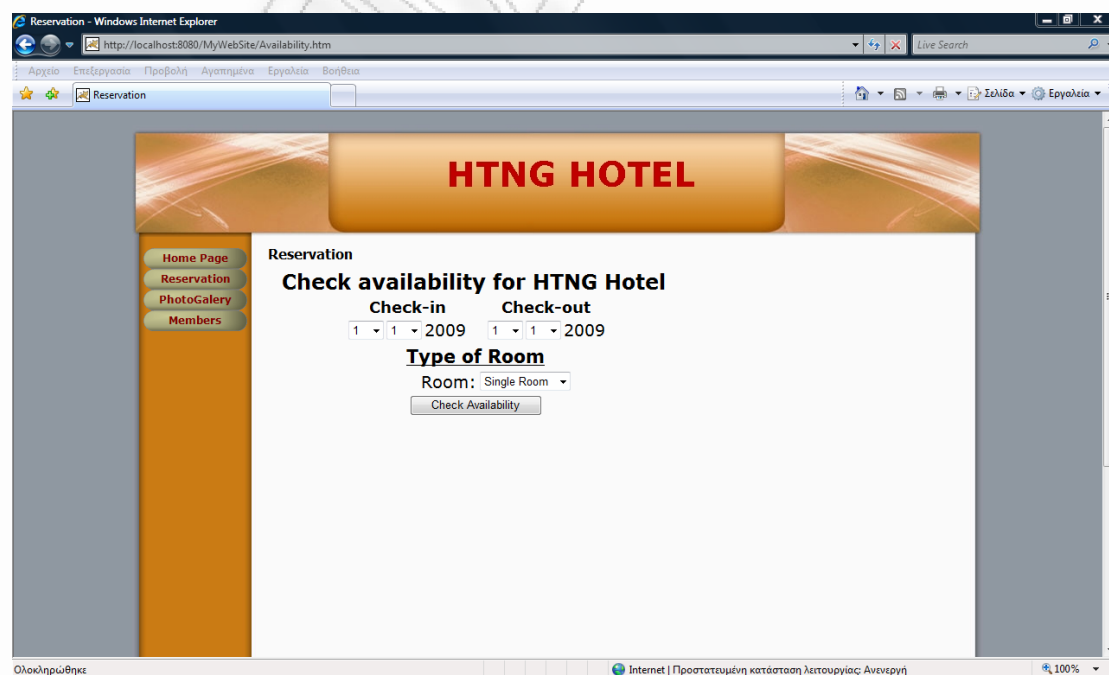
100%

Εδώ μπορούμε να δούμε γενικές πληροφορίες που αφορούν το ξενοδοχείο, όπως για παράδειγμα τη διεύθυνσή του, πληροφορίες σχετικά με το πώς θα φτάσουμε σε αυτό, πληροφορίες που αφορούν τον αριθμό των δωματίων του ξενοδοχείου, τις ώρες της ημέρας κατά τις οποίες γίνεται Check in και Check out, πληροφορίες για απόστάσεις μέσω αεροδρομίου, τρένου ή taxi, επίσης στη σελίδα αυτή θα βρούμε πληροφορίες σχετικά με τις υπηρεσίες οι οποίες παρέχονται αν υπάρχει περιοχή πρόσβασης στο Internet, εάν υπάρχει δυνατότητα Parking, ποια είναι η πολιτική του ξενοδοχείου σχετικά με τα παιδιά, εάν επιτρέπονται κατοικίδια, η δυνατότητα πληρωμής μέσω Credit Card.

Στην παρακάτω εικόνα μπορούμε να δούμε την σελίδα Reservation. Στην σελίδα αυτή μπορεί κάποιος μεμονωμένος χρήστης του Internet να επιλέξει την ημερομηνία που επιθυμεί καθώς και τον τύπο του δωματίου και πατώντας το κουμπί Check Availability να του εμφανιστεί η λίστα όλων των δωματίων εκείνων που τη συγκεκριμένη ημερομηνία είναι διαθέσιμα.

Οι επιλογές που μπορεί να κάνει όσον αφορά τον τύπο του δωματίου είναι:

- Single Room
- Double Room
- Triple Room
- Suite



Το αποτέλεσμα της αναζήτησης πχ. Από 1/1/2009 έως 2/1/2009 με τύπο δωματίου Single Room φαίνεται στην παρακάτω εικόνα:

RoomID	RoomName	RoomType	NumberOfBeds	Floor	TaxRateID	Unit Price	Com ID	Room Facilities	Selection
1	100	SGL	1	1	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
4	103	SGL	1	1	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
20	203	SGL	1	2	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
33	300	SGL	1	3	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
36	303	SGL	1	3	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
49	400	SGL	1	4	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
52	403	SGL	1	4	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
65	500	SGL	1	5	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
68	503	SGL	1	5	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
81	600	SGL	1	6	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
84	603	SGL	1	6	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select

Στην εικόνα αυτή μπορούμε να δούμε τις παρακάτω πληροφορίες τον κωδικό του δωματίου, το όνομα του δωματίου τον τύπο του δωματίου τον αριθμό των κρεβατιών τον όροφο που βρίσκεται το δωμάτιο, την τιμή του δωματίου και πληροφορίες σχετικά με το commission. Εάν για οποιοδήποτε λόγο διαπιστώσουμε ότι δεν μας ικανοποιεί η επιλογή μας αυτή τότε μπορούμε να πατήσουμε το κουμπί GoBack και να επιστρέψουμε στην προηγούμενη οθόνη για να επιλέξουμε εκ νέου με βάση κάποια άλλη ημερομηνία ή με άλλο τύπο δωματίου. Η επιλογή οποιοδήποτε από τα δωμάτια της λίστας μπορεί να γίνει πατώντας το κουμπί Select που βρίσκεται στο τέλος της κάθε γραμμής. Με το πάτημα του κουμπιού Select και την επιλογή κάποιου δωματίου περνάμε αυτόματα στην επόμενη σελίδα η οποία ονομάζεται ConfirmRoom.jsp.

Στη σελίδα αυτή μπορούμε να δούμε το δωμάτιο το οποίο έχουμε επιλέξει αλλά και να συνεχίσουμε τη διαδικασία προκειμένου να ολοκληρώσουμε την κράτηση μας με το να συμπληρώσουμε τα προσωπικά μας στοιχεία καθώς και στοιχεία για τον τύπο διατροφής που επιθυμούμε, εάν υπάρχει παιδί σύμφωνα με τον κανονισμό του συγκεκριμένου ξενοδοχείου για κάθε δωμάτιο υπάρχει η δυνατότητα φιλοξενίας ενός παιδιού μέχρι ηλικία δώδεκα χρόνων χωρίς χρέωση για κάθε δωμάτιο.

The screenshot shows a web browser window titled 'HTING HOTEL - Windows Internet Explorer'. The address bar shows the URL: <http://localhost:8080/MyWebSite/ConfirmRoom.jsp?RoomID=4&RoomName=103&RoomType=SGL&RoomNumBeds=1&RoomLoc=1&RoomTax=1&RoomPrice>. The page content is as follows:

You have Choose

RoomID:	4
Room Name:	103
Room Type:	SGL
Arrival Date:	2009/1/1
Depart Date:	2009/1/1
Adults:	1
Floor:	1
Price:	80.0 €
Booking Date:	2009-01-02

Personal Information

Guest Information

First Name:
Last Name:
Title: Mr
Home Address:
Work Address:
City:
State:
Zip Code:
Country:
Birth Day:
vip: No
Home Phone:
Work Phone:
FAX:
Email:

Credit Card Information

Credit Card Type: American Express - Advan
Card Number:
Expiration (MM/YY):
CVV2:

Comments

Additional Information

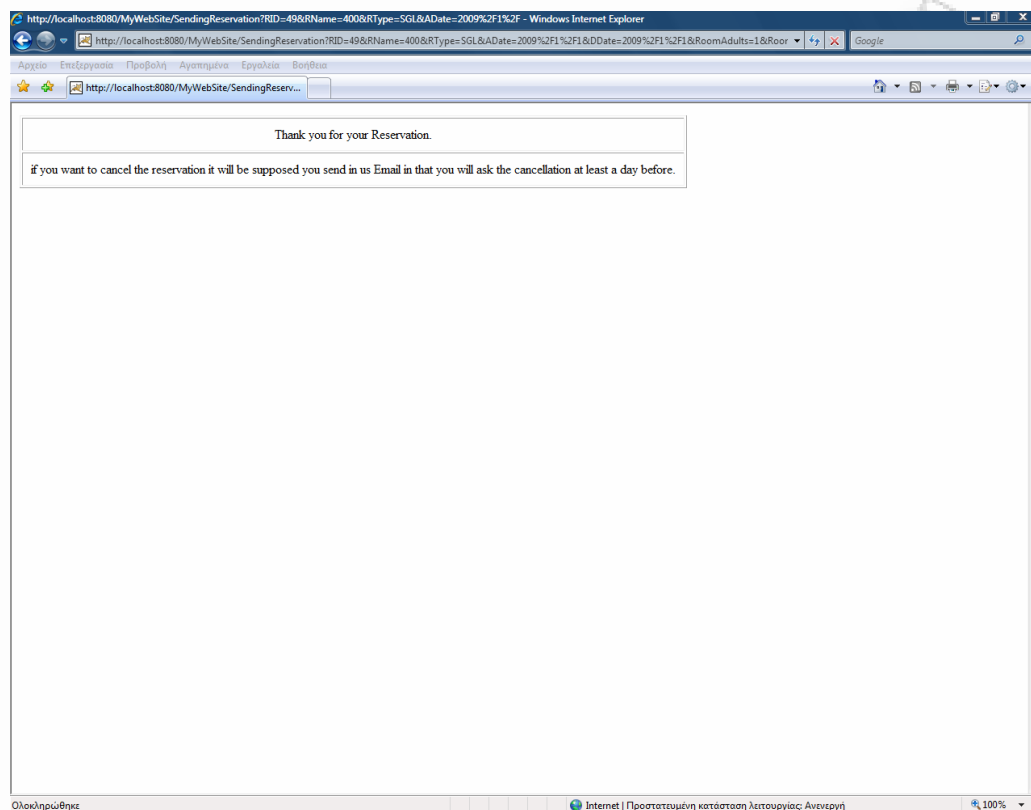
Meal: Room Rate
Child: No

Ολοκληρώθηκε

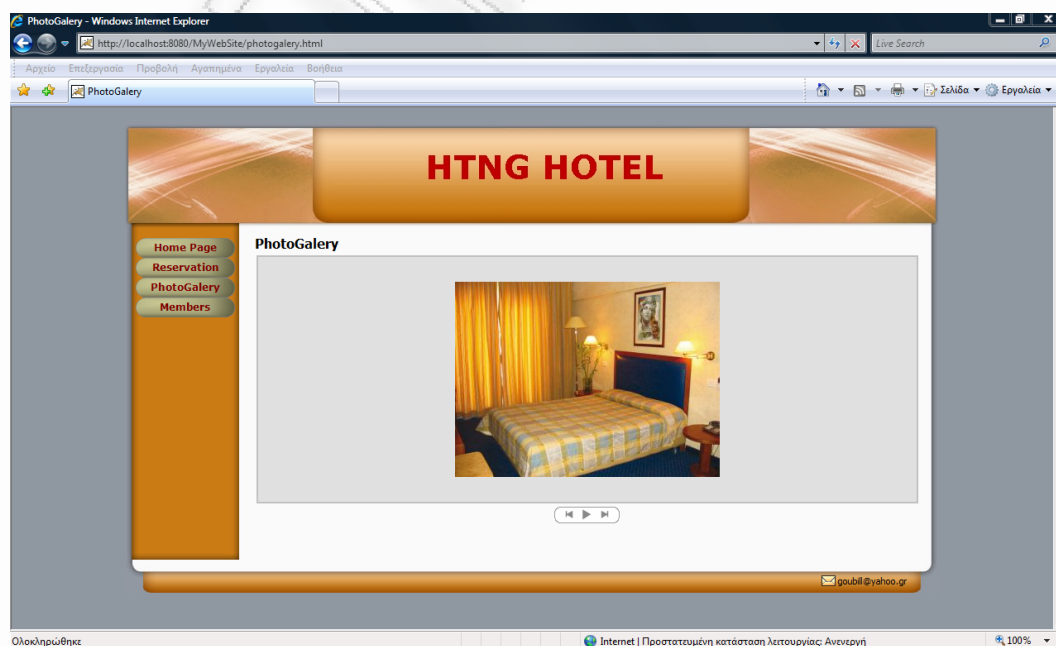
Internet | Προστατευμένη κατάσταση λειτουργίας: Ανενεργή

100%

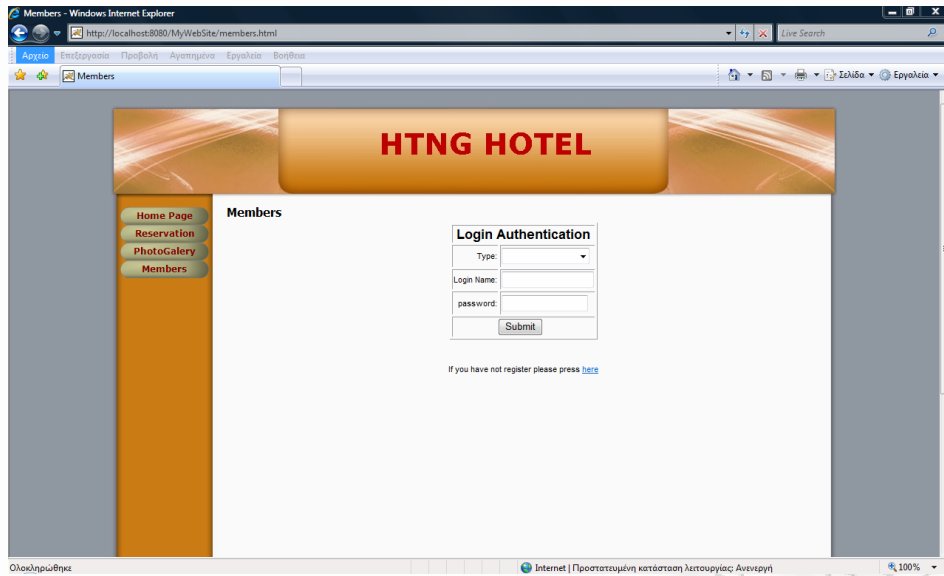
Επίσης όπως και προηγουμένως δίνεται η δυνατότητα να πάμε ένα βήμα πίσω για να επιλέξουμε κάποιο άλλο δωμάτιο με το κουμπί GoBack.



Από την αρχική σελίδα και μέσω του μενού επιλέγουμε PhotoGalery και στην οθόνη μας εμφανίζεται η παρακάτω εικόνα. Μέσω αυτής μπορούμε να έχουμε μια συγκεκριμένη άποψη για το πώς είναι διαμορφωμένα τα δωμάτια τη διακόσμηση, το χώρο του εστιατορίου, του bar και γενικότερα όλους τους χώρους που θέλει να αναδείξει ώστε ο πελάτης πηγαίνοντας εκεί να ξέρει τι ακριβώς θα βρεί.



Στην παρακάτω εικόνα βλέπουμε την σελίδα login.jsp



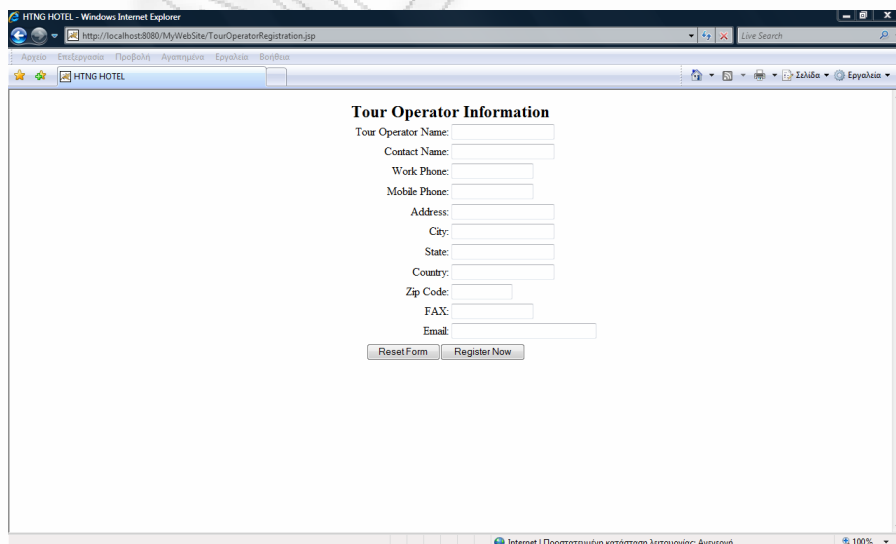
Στην σελίδα αυτή παρέχεται η δυνατότητα σε πιστοποιημένα άτομα όπως είναι οι tour operators ή υπαλλήλους του ξενοδοχείου να έχουν πρόσβαση και να εκτελούν συγκεκριμένες λειτουργίες. Υπάρχουν τρία πεδία τα οποία θα πρέπει να συμπληρωθούν (Type, LoginName, Password).

Το πεδίο Type μπορεί να πάρει τιμές:

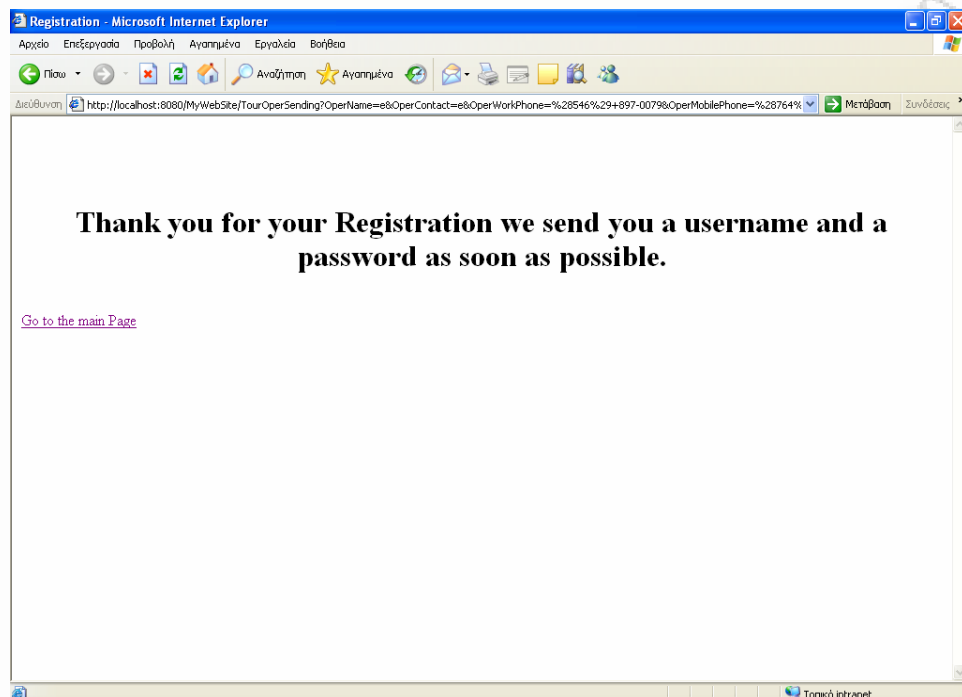
- Hotel Employee
- Tour Operator

Στην περίπτωση που ο tour operator μπαίνει για πρώτη φορά στην σελίδα θα πρέπει να κάνει registration όπου θα δηλώσει τα προσωπικά του στοιχεία έτσι ώστε να πάρει μέσω email από το ξενοδοχείο username και password ώστε να μπορεί να κάνει ομαδικές κρατήσεις.

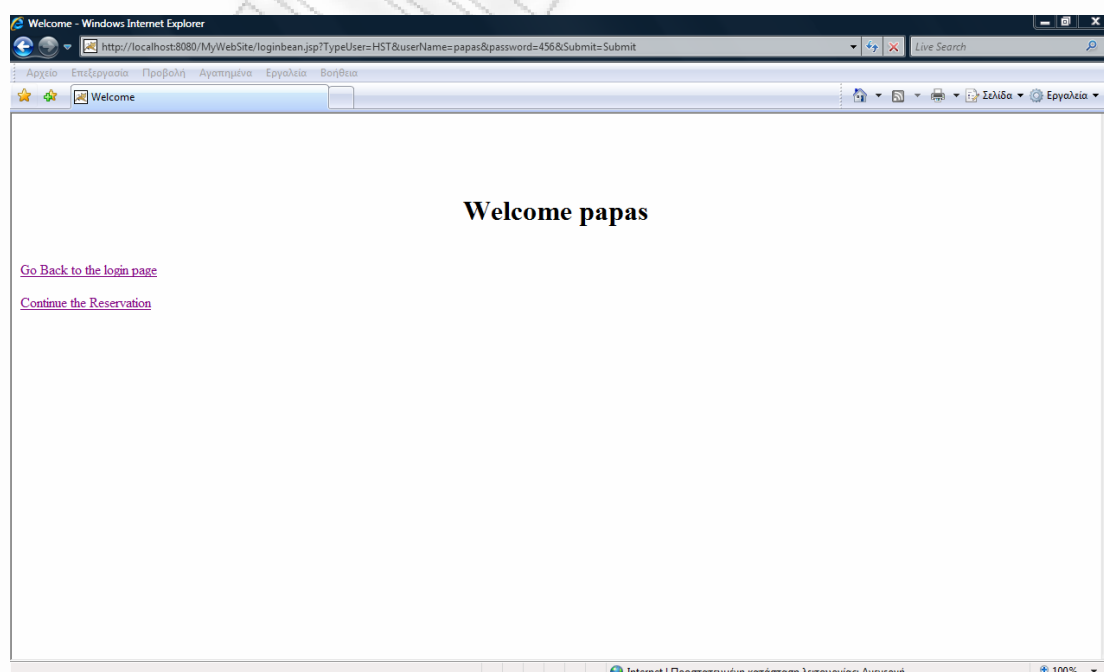
Στην παρακάτω εικόνα φαίνεται η φόρμα που θα πρέπει να συμπληρώσει ο tour operator ώστε να γίνει μέλος του ξενοδοχείου.



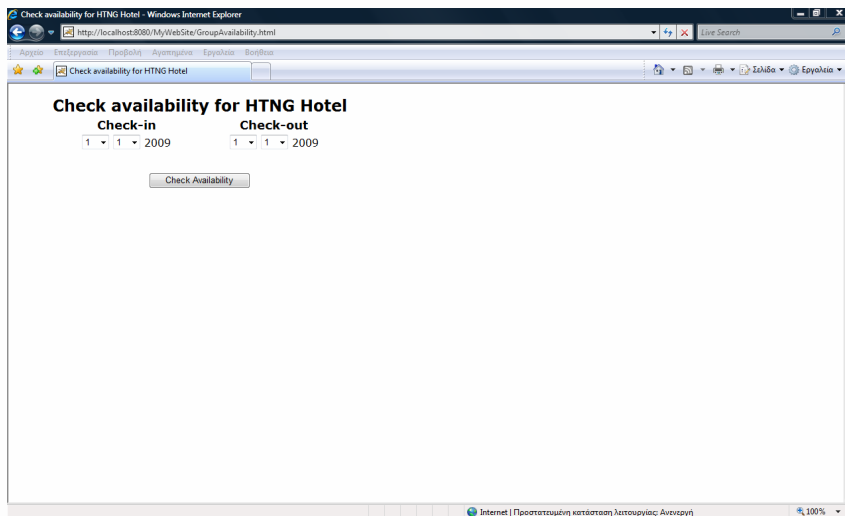
Πατώντας το κουμπί Register Now θα εμφανιστεί η παρακάτω οθόνη η οποία αφού ευχαριστήσει τον συγκεκριμένο tour operator για το registration τον ενημερώνει ότι θα του αποστείλει μέσω email το username και το password.



Έχοντας λοιπόν ο tour operator πιστοποιηθεί από το σύστημα και αφού εισάγει στην σελίδα το όνομα του χρήστη και τον κωδικό πρόσβασης θα μεταβεί στη σελίδα Welcome.jsp η οποία τον καλωσορίζει στο σύστημα και η οποία του δίνει τη δυνατότητα να συνεχίσει την διαδικασία κράτησης με το να επιλέξει το link "Continue the Reservation".



Ο παραπάνω σύνδεσμος μας οδηγεί στην σελίδα GroupAvailability.html.



Η σελίδα αυτή είναι σχεδόν όμοια με αυτή που χρησιμοποιείται για τους μεμονωμένους χρήστε που θέλουν να κάνουν Reservation όπως είδαμε παραπάνω. Εδώ μπορούμε με βάση την ημερομηνία να πάρουμε ολόκληρη τη λίστα των δωματίων του ξενοδοχείου που είναι διαθέσιμα.

Στην παρακάτω εικόνα φαίνεται η λίστα με τα διαθέσιμα δωμάτια του ξενοδοχείου χωρίς κατηγοριοποίηση.

RoomID	RoomName	RoomType	NumberOfBeds	Floor	TaxRateID	Unit Price	Com ID	Room Facilities	Selection
1	100	SGL	1	1	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
3	102	TPL	3	1	1	100.0	3	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
4	103	SGL	1	1	1	80.0	1	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
5	104	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
6	105	TPL	3	1	1	100.0	3	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
7	106	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
8	107	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
9	108	TPL	3	1	1	100.0	3	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
10	109	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
11	110	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
12	111	DBL	2	1	1	90.0	2	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select
13	112	TPL	3	1	1	100.0	3	Phone,Internet,Airconditioner,SatTV,In-roomService,Refrigerator	Select

Στο κάτω μέρος της οθόνης που εδώ δεν φαίνονται γιατί είναι πολύ μεγάλη σε μέγεθος υπάρχουν δύο κουμπιά το κουμπί Back και το κουμπί Next.

Το πρώτο μας γυρνά στην προηγούμενη οθόνη ενώ το επόμενο μας πάει στην επόμενη η οποία φαίνεται παρακάτω.

Η διαδικασία επιλογής περισσότερων του ενός δωματίου από αυτόν γίνεται ως εξής:

Από την προηγούμενη οθόνη στην οποία εμφανίζεται όλη η λίστα του ξενοδοχείου με τα διαθέσιμα δωμάτια επιλέγουμε πατώντας στο τέλος κάθε γραμμής το κουμπί Select. Αυτό έχει ως αποτέλεσμα να μας προωθήσει στην σελίδα GroupList.jsp στην οποία θα δείξει το τι έχουμε επιλέξει. Εάν επιθυμούμε να επιλέξουμε και άλλα δωμάτια πέραν του ήδη υπάρχοντος θα πάμε στο link “Add New Room” όπου θα μας πάει στην προηγούμενη σελίδα (στην λίστα) στην οποία δεν θα υπάρχει το δωμάτιο το οποίο έχουμε προηγούμενα επιλέξει και θα επιλέξουμε το επόμενο με τον ίδιο ακριβώς τρόπο που κάναμε και προηγουμένως κλπ.

Μόλις τελειώσουμε με την επιλογή των δωματίων (στην συγκεκριμένη περίπτωση έχω επιλέξει δύο δωμάτια το 107 και το 612 , το πρώτο είναι δίκλινο ενώ το δεύτερο τρίκλινο) θα πρέπει για να ολοκληρωθεί η διαδικασία της ομαδικής κράτησης να συμπληρώσει το username του tour operator να επιλέξει το είδος διατροφής που επιθυμεί και να δηλώσει αν υπάρχουν παιδιά.

Group Reservation

[Add New Room](#)

RoomID	Name	RoomType	Beds	Floor	TaxRateID	Unit Price	Com ID	Com Percent	Delete
8	107	DBL	2	1	1	90.0	2	2%	Delete
93	612	TPL	3	6	1	100.0	3	3%	Delete

Additional Information

Arrival Date:

Depart Date:

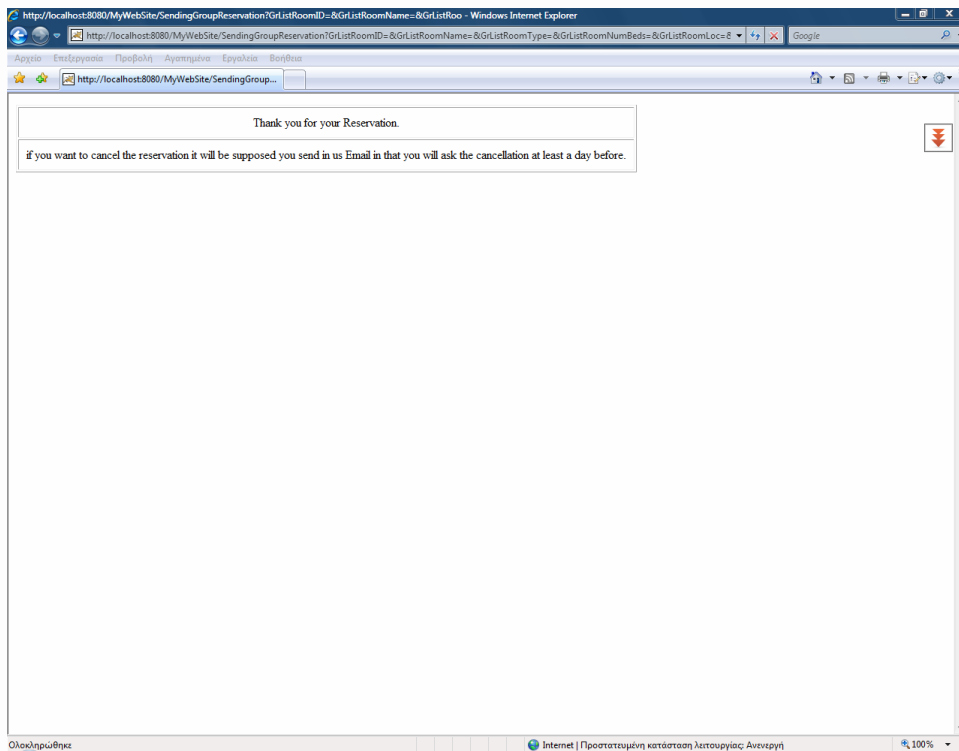
Booking Date:

UserName:

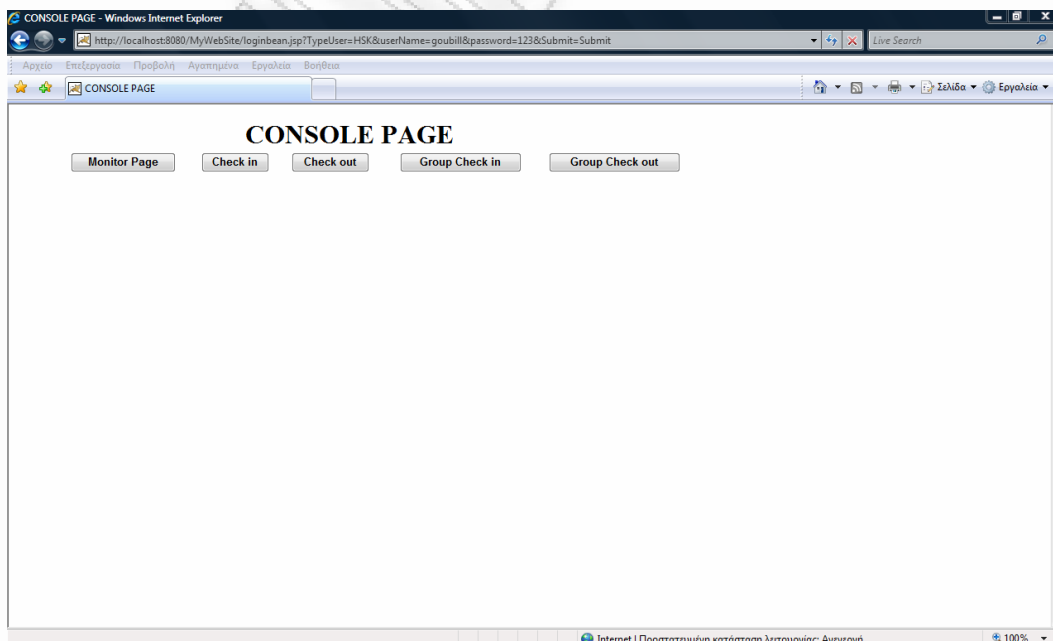
Meal:

Child:

Η κράτηση ολοκληρώνεται πατώντας το κουμπί Finish Reservation.



Τώρα περνάμε στην περίπτωση που κάποιος υπάλληλος του ξενοδοχείου κάνει login στο σύστημα τότε αυτό θα τον οδηγήσει στην σελίδα Console.jsp όπου βλέπουμε το πώς ακριβώς είναι.



Η σελίδα αυτή αποτελεί μια κονσόλα διαχείρισης με βάση την οποία ο υπάλληλος μπορεί να δει την τρέχουσα κατάσταση για κάποια συγκεκριμένη ημερομηνία όπως φαίνεται στην σελίδα Monitor.jsp στην οποία φαίνεται ότι τη συγκεκριμένη ημερομηνία έχει κλειστεί το δωμάτιο 106 αλλά δεν έχει γίνει Check in ενώ τα δωμάτια 101,200,500,503 και αφορούν την σημερινή ημερομηνία έχει γίνει Check in.

Όταν σε κάποιο δωμάτιο που εμφανίζεται ότι έχει γίνει Check in γίνει check out τότε επαναφέρει το χρώμα του σε λευκό και το διαγράφει από τον πίνακα που γίνονται οι κρατήσεις.

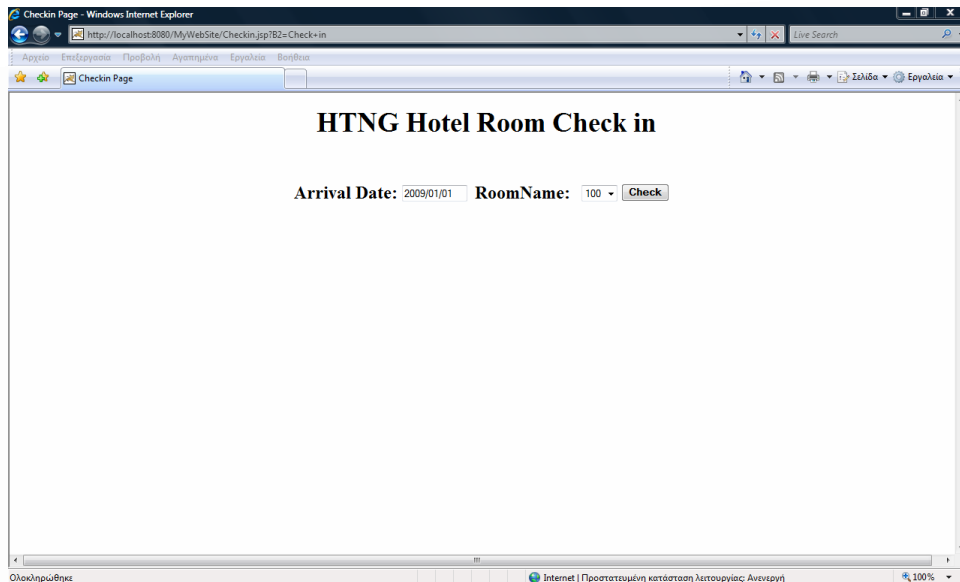
The screenshot shows a web browser window titled "Monitor Page - Microsoft Internet Explorer". The address bar shows the URL: `http://localhost:8080/MyWebSite/SendingMonitorServlet?Date=2009%2F1%2F1&CheckStatus=Check+Status`. The main content area is titled "Hotels Room Status Monitor" and features a "Select a Date:" field with a calendar icon and a "Check Status" button. Below this is a table of room statuses:

1st Floor	2nd Floor	3rd Floor	4th Floor	5th Floor	6th Floor	7th Floor						
100	108	200	208	300	308	400	408	500	508	600	608	Suite1
101	109	201	209	301	309	401	409	501	509	601	609	Suite2
102	110	202	210	302	310	402	410	502	510	602	610	Suite3
103	111	203	211	303	311	403	411	503	511	603	611	Suite4
104	112	204	212	304	312	404	412	504	512	604	612	
105	113	205	213	305	313	405	413	505	513	605	613	
106	114	206	214	306	314	406	414	506	514	606	614	
107	115	207	215	307	315	407	415	507	515	607	615	

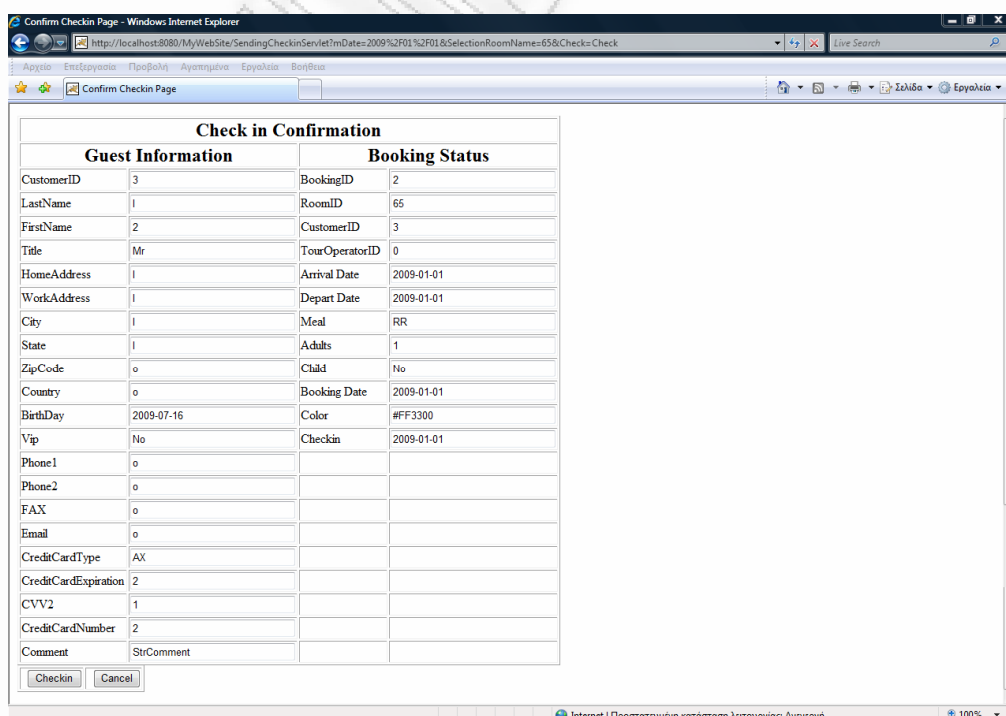
Legend: Vacant Booked Checked in

Buttons: Home Page, Console

Στην εικόνα αυτή εμφανίζεται η σελίδα Check in με την οποία μπορούμε να κάνουμε Check in επιλέγοντας το δωμάτιο, η ημερομηνία επιλέγεται αυτόματα από σύστημα.

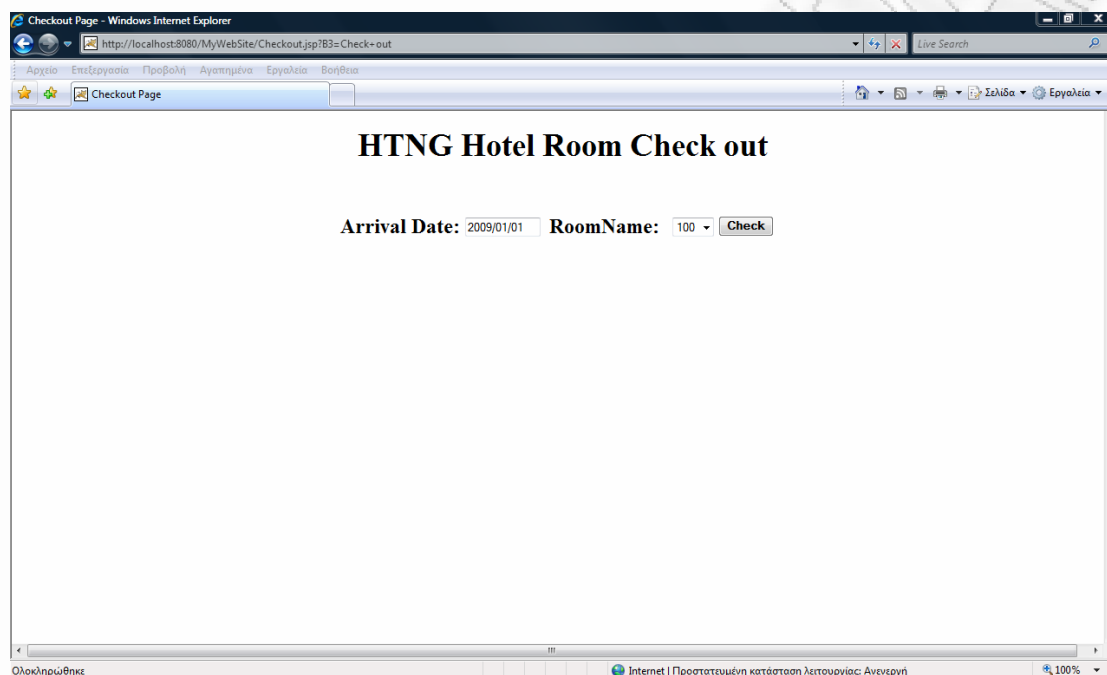


Πατώντας το κουμπί Check και αφού έχουμε επιλέξει το δωμάτιο που θέλουμε να κάνουμε Check in μας οδηγεί στη σελίδα ConfirmCheckin.jsp όπου βλέπουμε μια περιγραφή για το ποιος είναι αυτός ο οποίος έχει κλείσει το δωμάτιο όλα τα προσωπικά του στοιχεία καθώς και στοιχεία από τον πίνακα κρατήσεων όπως τον κωδικό του δωματίου, την ημερομηνία άφιξης, την ημερομηνία αναχώρησης καθώς και την ημερομηνία που έγινε η κράτηση. Αφού πιστοποιηθούν τα στοιχεία του πελάτη τότε ο υπάλληλος πατάει το κουμπί Checkin και ολοκληρώνει τη διαδικασία. Πηγαίνοντας στην σελίδα Monitor.jsp θα παρατηρήσουμε ότι το χρώμα θα αλλάξει από πράσινο σε κόκκινο το οποίο δηλώνει ότι στο συγκεκριμένο δωμάτιο έχει γίνει Checkin.

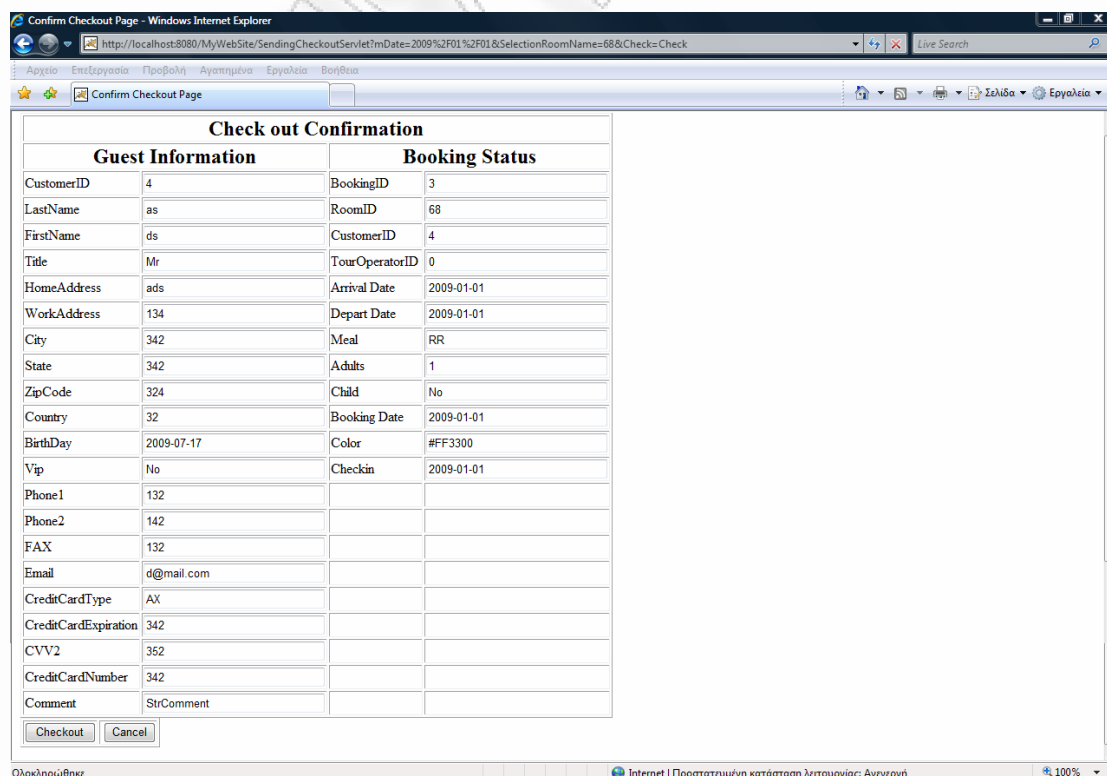


Η διαδικασία του Check out είναι ακριβώς η ίδια με τη διαφορά ότι όταν γίνεται σε ένα δωμάτιο τότε διαγράφεται η καταχώρηση στον πίνακα Booking οπότε η κατάσταση του δωματίου στην Console.jsp μεταβάλλεται αυτόματα σε διαθέσιμο.

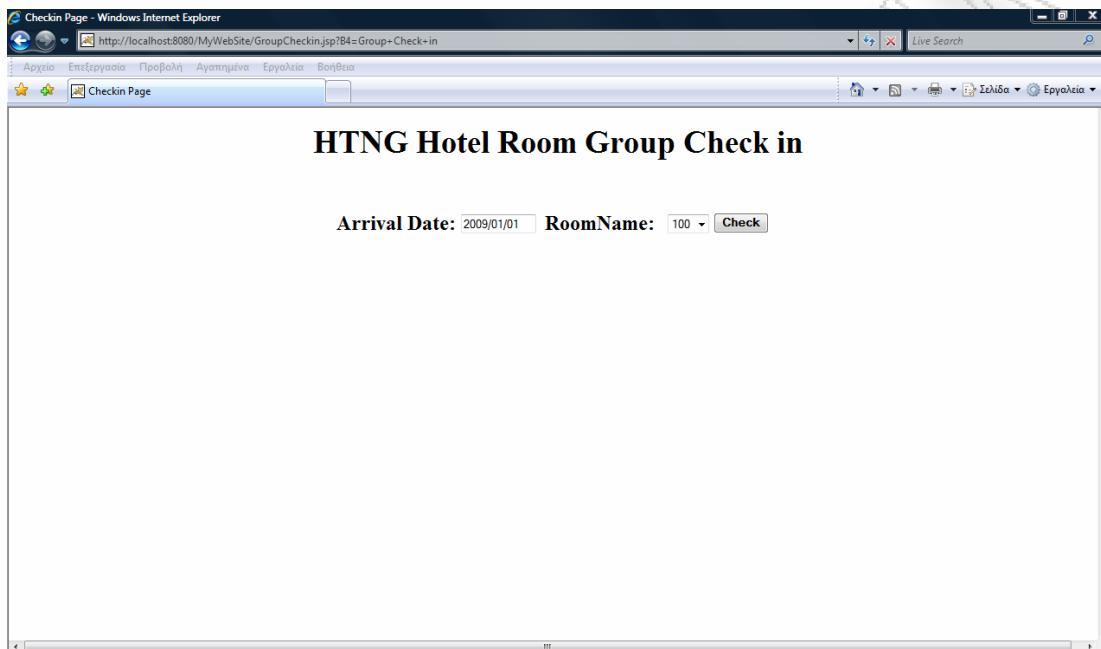
Παρακάτω βλέπουμε τις σχετικές σελίδες που έχουν να κάνουν με το Checkout.jsp και ConfirmCheckout.jsp. Η λειτουργία τους είναι όπως και παραπάνω



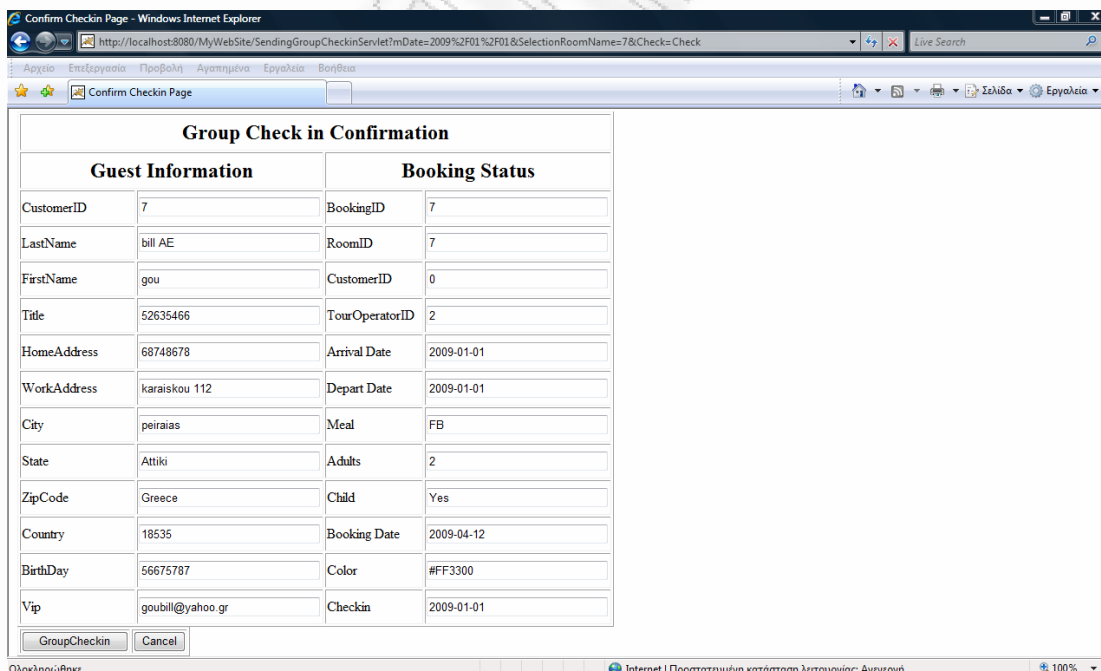
Η παρακάτω σελίδα κάνει Check out στο δωμάτιο με κωδικό 68.



Παρόμοια διαδικασία ακολουθείται και στην περίπτωση του Group Check in και του Group Check out, η μόνη διαφορά είναι ότι στην σελίδα που γίνεται η ταυτοποίηση πλέον αντί τα προσωπικά στοιχεία του πελάτη θα υπάρχουν τα προσωπικά στοιχεία του tour operator.



Στην παρακάτω σελίδα γίνεται Check in του δωματίου με κωδικό 7.



Παρόμοια διαδικασία για Group Check out

Checkout Page - Windows Internet Explorer
http://localhost:8080/MyWebSite/GroupCheckout.jsp?85=Group+Check+out

Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια

Checkout Page

HTNG Hotel Room Group Check out

Arrival Date: 2009/01/01 RoomName: 100

Ολοκληρώθηκε

Internet | Προστατευμένη κατάσταση λειτουργίας: Ανενεργή 100%

Στην παρακάτω σελίδα γίνεται Check out του δωματίου με κωδικό 7.

Confirm Checkout Page - Windows Internet Explorer
http://localhost:8080/MyWebSite/SendingGroupCheckoutServlet?mDate=2009%2F01%2F01&SelectionRoomNames=7&Check=Check

Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια

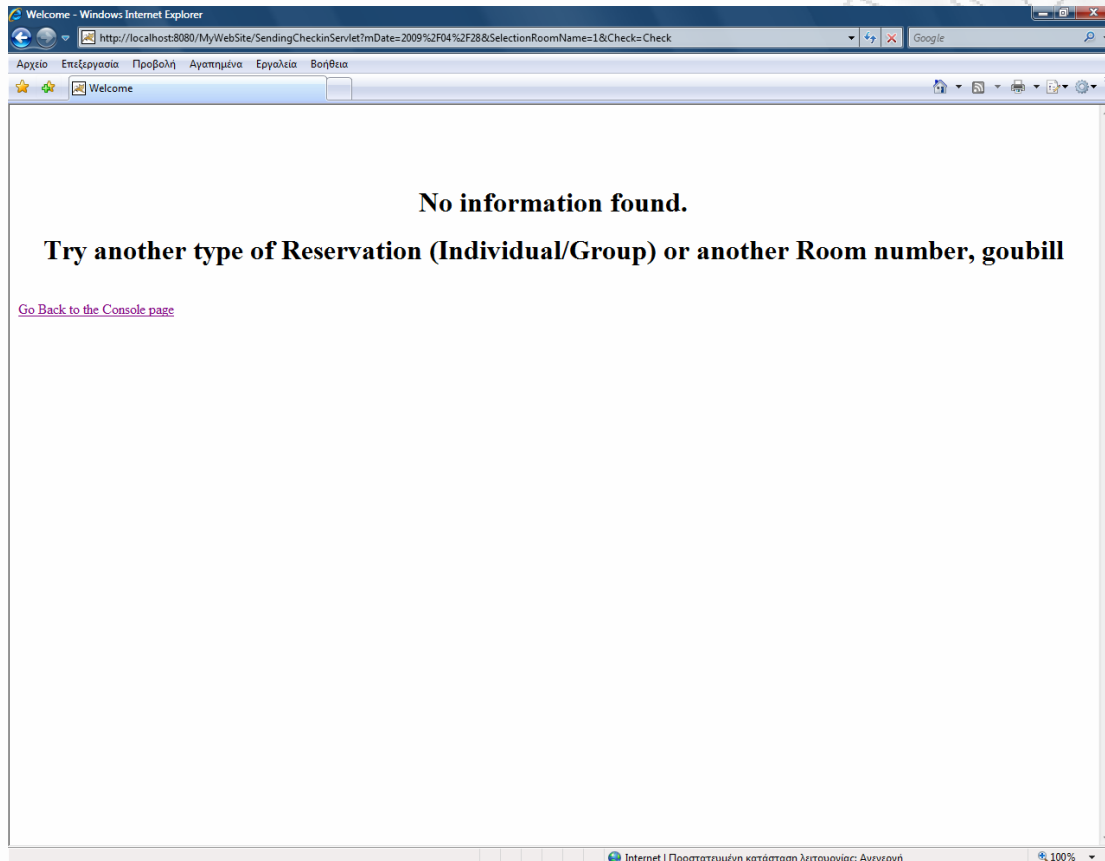
Confirm Checkout Page

Group Check out Confirmation			
Guest Information		Booking Status	
CustomerID	7	BookingID	7
LastName	bill AE	RoomID	7
FirstName	gou	CustomerID	0
Title	52635466	TourOperatorID	2
HomeAddress	68748678	Arrival Date	2009-01-01
WorkAddress	karaiskou 112	Depart Date	2009-01-01
City	peiraias	Meal	FB
State	Attiki	Adults	2
ZipCode	Greece	Child	Yes
Country	18535	Booking Date	2009-04-12
BirthDay	56675787	Color	#FF3300
Vip	goubill@yahoo.gr	Checkin	2009-01-01

Ολοκληρώθηκε

Internet | Προστατευμένη κατάσταση λειτουργίας: Ανενεργή 100%

Στην περίπτωση που επιλεγεί κάποιο δωμάτιο το οποίο δεν έχει γίνει Book τότε εμφανίζεται η παρακάτω σελίδα η οποία μας προτρέπει να επιλέξουμε κάποιο άλλο δωμάτιο. Η ίδια σελίδα θα εμφανιστεί για παράδειγμα και εάν προσπαθήσουμε να κάνουμε Check in κάποιο δωμάτιο το οποίο αφορά Group αλλά και το αντίστροφο.



XML Μηνύματα της εφαρμογής

SendingServlet.java

AvailabilityRequest.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">AvailabilityRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:ADay xmlns:soap="http://htng.org">2009/1/1</soap:ADay>
<soap:DDay xmlns:soap="http://htng.org">2009/1/2</soap:DDay>
<soap:SelRoom xmlns:soap="http://htng.org">SGL</soap:SelRoom>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ReceivingServlet.java

AvailabilityResponse.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:AvailabilityResponse
xmlns:soap="http://htng.org">OK</soap:AvailabilityResponse>
<soap:RoomTable xmlns:soap="http://htng.org"><RoomID>1</RoomID>
<RoomName>100</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>1</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>4</RoomID>
<RoomName>103</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>1</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>17</RoomID>
<RoomName>200</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>2</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>20</RoomID>
<RoomName>203</RoomName>
```

```
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>2</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>33</RoomID>
<RoomName>300</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>3</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>36</RoomID>
<RoomName>303</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>3</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>49</RoomID>
<RoomName>400</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>4</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>52</RoomID>
<RoomName>403</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>4</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>65</RoomID>
<RoomName>500</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>5</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>68</RoomID>
<RoomName>503</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>5</Location>
```

```

<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>81</RoomID>
<RoomName>600</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>6</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
<RoomID>84</RoomID>
<RoomName>603</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>6</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
</soap:RoomTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingReservation.java

ReservationRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">ReservationRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap>Last xmlns:soap="http://htng.org">gourgos</soap>Last>
<soap:First xmlns:soap="http://htng.org">bill</soap:First>
<soap>Title xmlns:soap="http://htng.org">Mr</soap>Title>
<soap:Address1 xmlns:soap="http://htng.org">Karagiorgi
25</soap:Address1>
<soap:Address2 xmlns:soap="http://htng.org">Amalias
12</soap:Address2>
<soap:City xmlns:soap="http://htng.org">Athens</soap:City>
<soap:State xmlns:soap="http://htng.org"/>
<soap:Zip xmlns:soap="http://htng.org">18535</soap:Zip>
<soap:Country xmlns:soap="http://htng.org">Greece</soap:Country>
<soap:Birthday
xmlns:soap="http://htng.org">1972/07/18</soap:Birthday>
<soap:Vip xmlns:soap="http://htng.org">No</soap:Vip>
<soap:Phone1 xmlns:soap="http://htng.org">(210) 783-
4774</soap:Phone1>
<soap:Phone2 xmlns:soap="http://htng.org">(210) 978-
4394</soap:Phone2>
<soap:Fax xmlns:soap="http://htng.org">(210) 854-7985</soap:Fax>

```

```

<soap:Email
xmlns:soap="http://htng.org">gourgosvasilios@mintour.gr</soap:Email>
<soap:CreditCardType
xmlns:soap="http://htng.org">AX</soap:CreditCardType>
<soap:CardNumber
xmlns:soap="http://htng.org">42365867</soap:CardNumber>
<soap:Expiration xmlns:soap="http://htng.org">09/23</soap:Expiration>
<soap:CVV2 xmlns:soap="http://htng.org">43675548</soap:CVV2>
<soap:Comment xmlns:soap="http://htng.org"/>
<soap:Meal xmlns:soap="http://htng.org">FB</soap:Meal>
<soap:Child xmlns:soap="http://htng.org">No</soap:Child>
<soap:RID xmlns:soap="http://htng.org">98</soap:RID>
<soap:ADate xmlns:soap="http://htng.org">2009/1/1</soap:ADate>
<soap:DDate xmlns:soap="http://htng.org">2009/1/2</soap:DDate>
<soap:RoomAdults xmlns:soap="http://htng.org">4</soap:RoomAdults>
<soap:BDate xmlns:soap="http://htng.org">2009-04-27</soap:BDate>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingReservation.java

ReservationResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:ReservationResponse
xmlns:soap="http://htng.org">OK</soap:ReservationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingLogin.java

LoginRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">LoginRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body><soap:typeuser
xmlns:soap="http://htng.org">HSK</soap:typeuser>
<soap:uname xmlns:soap="http://htng.org">goubill</soap:uname>
<soap:upass xmlns:soap="http://htng.org">123</soap:upass>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingLogin.java

LoginResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

```

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:LoginResponse
xmlns:soap="http://htng.org">OK</soap:LoginResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingMonitorServlet.java

MonitorRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">MonitorRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:MyDate xmlns:soap="http://htng.org">2009/1/1</soap:MyDate>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingMonitorServlet.java

MonitorResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:MonitorResponse
xmlns:soap="http://htng.org">OK</soap:MonitorResponse>
<soap:StatusTable xmlns:soap="http://htng.org">
<RoomID>65</RoomID>
<Color>#FF3300</Color>
<RoomID>68</RoomID>
<Color>#FF3300</Color>
<RoomID>17</RoomID>
<Color>#FF3300</Color>
<RoomID>7</RoomID>
<Color>#FF3300</Color>
<RoomID>1</RoomID>
<Color>#009900</Color>
<RoomID>2</RoomID>
<Color>#009900</Color>
<RoomID>23</RoomID>
<Color>#009900</Color>
<RoomID>2</RoomID>
<Color>#009900</Color>
<RoomID>1</RoomID>
<Color>#009900</Color>
<RoomID>2</RoomID>
<Color>#009900</Color>
<RoomID>23</RoomID>
<Color>#009900</Color>
<RoomID>2</RoomID>
<Color>#009900</Color>
<RoomID>4</RoomID>
<Color>#009900</Color>

```



```
<RoomID>13</RoomID>
<Color>#009900</Color>
</soap:StatusTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

TourOperSending.java

TourOperatorRequest.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">TourOperatorRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:OperName xmlns:soap="http://htng.org">FGK AE</soap:OperName>
<soap:OperContact xmlns:soap="http://htng.org">bill
gourgos</soap:OperContact>
<soap:OperWorkPhone xmlns:soap="http://htng.org">(210) 948-
3539</soap:OperWorkPhone>
<soap:OperMobilePhone xmlns:soap="http://htng.org">(694) 457-
8698</soap:OperMobilePhone>
<soap:OperAddress xmlns:soap="http://htng.org">xormovitou
345</soap:OperAddress>
<soap:OperCity xmlns:soap="http://htng.org">Athens</soap:OperCity>
<soap:OperState xmlns:soap="http://htng.org">Athens</soap:OperState>
<soap:OperCountry
xmlns:soap="http://htng.org">Greece</soap:OperCountry>
<soap:OperZipCode
xmlns:soap="http://htng.org">18535</soap:OperZipCode>
<soap:OperFAX xmlns:soap="http://htng.org">(210) 483-
7786</soap:OperFAX>
<soap:OperEmail
xmlns:soap="http://htng.org">gourgosvasilios@mintour.gr</soap:OperEma
il>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

TourOperReceiving.java

TourOperatorResponse.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:TourOperatorResponse
xmlns:soap="http://htng.org">OK</soap:TourOperatorResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SendingGroupServlet.java

GroupAvailabilityRequest.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

```

    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">GroupAvailabilityRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:ADay xmlns:soap="http://htng.org">2009/1/3</soap:ADay>
<soap:DDay xmlns:soap="http://htng.org">2009/1/4</soap:DDay>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingGroupServlet.java

GroupAvailabilityResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:GroupAvailabilityResponse
xmlns:soap="http://htng.org">OK</soap:GroupAvailabilityResponse>
<soap:RoomTable xmlns:soap="http://htng.org">
<RoomID>1</RoomID>
<RoomName>100</RoomName>
<RoomType>SGL</RoomType>
<NumOfBeds>1</NumOfBeds>
<Location>1</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>80.0</UnitPrice>
<CommissionID>1</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
.....
.....
.....
.....

<RoomID>100</RoomID>
<RoomName>S4</RoomName>
<RoomType>SUITE</RoomType>
<NumOfBeds>4</NumOfBeds>
<Location>7</Location>
<TaxRateID>1</TaxRateID>
<UnitPrice>200.0</UnitPrice>
<CommissionID>4</CommissionID>
<RoomFacilities>Phone, Internet, Airconditioner, SatTV, In-
roomService, Refrigerator</RoomFacilities>
</soap:RoomTable></SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingGroupReservation.java

GroupReservationRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

```

<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">GroupReservationRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:Meal xmlns:soap="http://htng.org">HB</soap:Meal>
<soap:gChild xmlns:soap="http://htng.org">No</soap:gChild>
<soap:gArrDay xmlns:soap="http://htng.org">2009/1/5</soap:gArrDay>
<soap:gDepDay xmlns:soap="http://htng.org">2009/1/6</soap:gDepDay>
<soap:gDateNew
xmlns:soap="http://htng.org">2009/04/27</soap:gDateNew>
<soap:gUserName xmlns:soap="http://htng.org">papas</soap:gUserName>
<soap:ListSize xmlns:soap="http://htng.org">18</soap:ListSize>
<soap:Room1 xmlns:soap="http://htng.org">8</soap:Room1>
<soap:Adults1 xmlns:soap="http://htng.org">2</soap:Adults1>
<soap:Room2 xmlns:soap="http://htng.org">13</soap:Room2>
<soap:Adults2 xmlns:soap="http://htng.org">3</soap:Adults2>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingGroupReservation.java

GroupReservationResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:GroupReservationResponse
xmlns:soap="http://htng.org">OK</soap:GroupReservationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingCheckinServlet.java

ConfirmCheckinRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">ConfirmCheckinRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:MyDate xmlns:soap="http://htng.org">2009/01/01</soap:MyDate>
<soap:StrRoomID xmlns:soap="http://htng.org">68</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingCheckinServlet.java

ConfirmCheckinResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>

```

```

<soap:ConfirmCheckinResponse
xmlns:soap="http://htng.org">OK</soap:ConfirmCheckinResponse>
<soap:StatusTable xmlns:soap="http://htng.org">
<BookingID>3</BookingID>
<RoomID>68</RoomID>
<CustomerID>4</CustomerID>
<TourOperatorID>0</TourOperatorID>
<ArrivalDate>2009-01-01</ArrivalDate>
<DepartDate>2009-01-01</DepartDate>
<Meal>RR</Meal>
<Adults>1</Adults>
<Child>No</Child>
<BookingDate>2009-01-01</BookingDate>
<Color>#FF3300</Color>
<Checkin>2009-01-01</Checkin>
<customer_id>4</customer_id>
<LastName>Gourgos</LastName>
<FirstName>bill</FirstName>
<Title>Mr</Title>
<HomeAddress>karaiskou 112</HomeAddress>
<WorkAddress>Amalias 12</WorkAddress>
<City>Athens</City>
<State>Athens</State>
<ZipCode>32445</ZipCode>
<Country>Greece</Country>
<BirthDay>2009-07-17</BirthDay>
<Vip>No</Vip>
<Phone1></Phone1>
<Phone2></Phone2>
<Fax></Fax><Email>d@mail.com</Email>
<CreditCardType>AX</CreditCardType>
<CreditCardExpiration></CreditCardExpiration>
<CVV2></CVV2>
<CreditCardNumber></CreditCardNumber>
<Comment></Comment>
</soap:StatusTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendCheckinServlet.java

CheckinRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">CheckinRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:StrRoomID xmlns:soap="http://htng.org">68</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivCheckinServlet.java

CheckinResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

```

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:CheckinResponse
xmlns:soap="http://htng.org">OK</soap:CheckinResponse>
<soap:StatusTable xmlns:soap="http://htng.org"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingCheckoutServlet.java

ConfirmCheckoutRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">ConfirmCheckoutRequest</soap:MessageName
>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:MyDate xmlns:soap="http://htng.org">2009/01/01</soap:MyDate>
<soap:StrRoomID xmlns:soap="http://htng.org">65</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivingCheckoutServlet.java

ConfirmCheckoutResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:ConfirmCheckoutResponse
xmlns:soap="http://htng.org">OK</soap:ConfirmCheckoutResponse>
<soap:StatusTable xmlns:soap="http://htng.org">
<BookingID>2</BookingID>
<RoomID>65</RoomID>
<CustomerID>3</CustomerID>
<TourOperatorID>0</TourOperatorID>
<ArrivalDate>2009-01-01</ArrivalDate>
<DepartDate>2009-01-01</DepartDate>
<Meal>RR</Meal>
<Adults>1</Adults>
<Child>No</Child>
<BookingDate>2009-01-01</BookingDate>
<Color>#FF3300</Color>
<Checkin>2009-01-01</Checkin>
<customer_id></customer_id>
<LastName></LastName>
<FirstName></FirstName>
<Title>Mr</Title>
<HomeAddress></HomeAddress>
<WorkAddress></WorkAddress>
<City></City>
<State></State>
<ZipCode></ZipCode>
<Country></Country>
<BirthDay>2009-07-16</BirthDay>
<Vip>No</Vip>

```

```

<Phone1></Phone1>
<Phone2></Phone2>
<Fax></Fax>
<Email></Email>
<CreditCardType>AX</CreditCardType>
<CreditCardExpiration></CreditCardExpiration>
<CVV2></CVV2>
<CreditCardNumber></CreditCardNumber>
<Comment></Comment>
</soap:StatusTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendCheckoutServlet.java

CheckoutRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">CheckoutRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:StrRoomID xmlns:soap="http://htng.org">65</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivCheckoutServlet.java

CheckoutResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:CheckoutResponse
xmlns:soap="http://htng.org">OK</soap:CheckoutResponse>
<soap:StatusTable xmlns:soap="http://htng.org"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendingGroupCheckinServlet.java

ConfirmGroupCheckinRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">ConfirmGroupCheckinRequest</soap:Message
Name>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:MyDate xmlns:soap="http://htng.org">2009/01/01</soap:MyDate>
<soap:StrRoomID xmlns:soap="http://htng.org">7</soap:StrRoomID>

```

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ReceivingGroupCheckinServlet.java

ConfirmGroupCheckinResponse.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:ConfirmGroupCheckinResponse
xmlns:soap="http://htng.org">OK</soap:ConfirmGroupCheckinResponse>
<soap:StatusTable xmlns:soap="http://htng.org">
<BookingID>7</BookingID>
<RoomID>7</RoomID>
<CustomerID>0</CustomerID>
<TourOperatorID>2</TourOperatorID>
<ArrivalDate>2009-01-01</ArrivalDate>
<DepartDate>2009-01-01</DepartDate>
<Meal>FB</Meal>
<Adults>2</Adults>
<Child>Yes</Child>
<BookingDate>2009-04-12</BookingDate>
<Color>#FF3300</Color>
<Checkin>2009-01-01</Checkin>
<ID>7</ID>
<OperatorName>bill AE</OperatorName>
<ContactName>gou</ContactName>
<WorkPhone>52635466</WorkPhone>
<MobilePhone>68748678</MobilePhone>
<Address>karaiskou 112</Address>
<City>peiraias</City>
<State>Attiki</State>
<Country>Greece</Country>
<ZipCode>18535</ZipCode>
<FAX>56675787</FAX>
<Email>goubill@yahoo.gr</Email>
</soap:StatusTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SendGroupCheckinServlet.java

GroupCheckinRequest.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">GroupCheckinRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:StrRoomID xmlns:soap="http://htng.org">7</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ReceivGroupCheckinServlet.java

GroupCheckinResponse.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:GroupCheckinResponse
xmlns:soap="http://htng.org">OK</soap:GroupCheckinResponse>
<soap:StatusTable xmlns:soap="http://htng.org"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SendingGroupCheckoutServlet.java

ConfirmGroupCheckoutRequest.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">ConfirmGroupCheckoutRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:MyDate xmlns:soap="http://htng.org">2009/01/01</soap:MyDate>
<soap:StrRoomID xmlns:soap="http://htng.org">7</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ReceivingGroupCheckoutServlet.java

ConfirmGroupCheckoutResponse.msg

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:ConfirmGroupCheckoutResponse
xmlns:soap="http://htng.org">OK</soap:ConfirmGroupCheckoutResponse>
<soap:StatusTable xmlns:soap="http://htng.org">
<BookingID>7</BookingID>
<RoomID>7</RoomID>
<CustomerID>0</CustomerID>
<TourOperatorID>2</TourOperatorID>
<ArrivalDate>2009-01-01</ArrivalDate>
<DepartDate>2009-01-01</DepartDate>
<Meal>FB</Meal>
<Adults>2</Adults>
<Child>Yes</Child>
<BookingDate>2009-04-12</BookingDate>
<Color>#FF3300</Color>
<Checkin>2009-01-01</Checkin>
<ID>7</ID>
<OperatorName>bill AE</OperatorName>
<ContactName>gou</ContactName>
```



```

<WorkPhone>52635466</WorkPhone>
<MobilePhone>68748678</MobilePhone>
<Address>karaiskou 112</Address>
<City>peiraias</City>
<State>Attiki</State>
<Country>Greece</Country>
<ZipCode>18535</ZipCode>
<FAX>56675787</FAX>
<Email>goubill@yahoo.gr</Email>
</soap:StatusTable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SendGroupCheckoutServlet.java

GroupCheckoutRequest.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
<soap:MessageName
xmlns:soap="http://htng.org">GroupCheckoutRequest</soap:MessageName>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<soap:StrRoomID xmlns:soap="http://htng.org">7</soap:StrRoomID>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ReceivGroupCheckoutServlet.java

GroupCheckoutResponse.msg

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<soap:GroupCheckoutResponse
xmlns:soap="http://htng.org">OK</soap:GroupCheckoutResponse>
<soap:StatusTable xmlns:soap="http://htng.org"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Επέκταση των δυνατοτήτων του προγράμματος

- Λειτουργίες που αφορούν τη διαχείριση των υπαλλήλων, δηλαδή δυνατότητα εισαγωγής, τροποποίησης και διαγραφής.
- Διατήρηση ιστορικού καταχωρήσεων των κρατήσεων που έχουν πραγματοποιηθεί έως σήμερα.
- Αναφορικά με την ομαδική κράτηση, δυνατότητα εισαγωγής στοιχείων του πελάτη κατά τη διαδικασία του check in.
- Στατιστικά στοιχεία που αφορούν στο ποσοστό κρατήσεων δωματίων κάποιας συγκεκριμένης ημερομηνίας ή ενός συγκεκριμένου χρονικού διαστήματος.
- Εμφάνιση υπό μορφή δισδιάστατου πίνακα της κατάστασης των κρατήσεων για μια συγκεκριμένη χρονική περίοδο, ανά μήνα, ανά σεζόν κλπ
- Επιλογή και χρέωση υπηρεσιών
- Υπολογισμός συνολικού κόστους διαμονής, συμπεριλαμβανομένων των παροχών υπηρεσιών εκτός δωματίου
- Έκδοση τιμολογίου ή απόδειξης.
- Δυνατότητα επιλογής κάποιου συγκεκριμένου πακέτου προσφοράς.
- Διαφορετική τιμολόγηση δωματίου ανάλογα με την περίοδο κράτησης.

Βιβλιογραφία και διευθύνσεις στο Internet

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ INTERNET ΜΕ ΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ JAVA SERVLETS (Κακαρόντζας Γιώργος)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ INTERNET ΜΕ ΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ JAVA SERVER PAGES (JSP) (Α' Μέρος) (Κακαρόντζας Γιώργος)

Julie C. Melony «PHP, MySql και Apache» Εκδόσεις Μ.Γκιούρδας

<http://www.filaderlis.com/ebooks/servlets.pdf>

<http://artemis.cslab.ntua.gr/Dienst/UI/1.0/Download/artemis.ntua.ece/DT2003-0160>

http://www.ted.unipi.gr/Uploads/Files/Material/Courses/160_1227814771.pdf

<http://web.auth.gr/virtualschool/Service/WhatIsInternet.html>

<http://www.noc.teikav.edu.gr/gr/1024x768/help/www.htm>

<http://www.it.uom.gr/project/soap/Theory/SOAP.html>

http://www2.aegean.gr/dgavalas/DP_I/slides/DP_I_2006_07.pdf

http://www.cs.ucy.ac.cy/~ep1425/tutorials/java_servlets.pdf

<http://www.it.uom.gr/project/soap/Downloads/Presentation.ppt>

<http://wim.aueb.gr/iraklis/courses/java/18%20-%20Java%20and%20XML.pdf>

http://ecourses.dbnet.ntua.gr/fsr/16335/1a_Introductory_Lecture.ppt

<http://dev.mysql.com/doc/refman/5.1/en/history.html>

<http://www.mysql.com/why-mysql/>

<http://www.clearlearning.com/v4/install/mysql.html>

<http://en.wikipedia.org/wiki/MySql>

http://en.wikipedia.org/wiki/Apache_HTTP_Server

<http://httpd.apache.org/>

<http://en.wikipedia.org/wiki/PhpMyAdmin>

http://www.htng.org/workgroups/irt/IRT_WhitePapers.htm

http://nefeli.lib.teicrete.gr/browse/stef/epp/2007/Vafeas,Christos/document/Anafora_ptyxiakis.pdf