Τ. Ε. Ι. Κρήτης

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πληροφοριακό σύστημα για ανταλλακτικά αυτοκινήτων (αποθήκη) με χρήση Java & Servlets

Κεντρής Αλέξανδρος Α.Μ. 2746

Σοφίου Ευάγγελος Α.Μ. 1916

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ

- 2015 -

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει σα σκοπό την υλοποίηση ενός Πληροφοριακού συστήματος για ανταλλακτικά αυτοκινήτων (αποθήκη) με χρήση Java & Servlets. Το πληροφοριακό σύστημα θα έχει τη δυνατότητα να αποθηκεύει πληροφορίες για προϊόντα, προμηθευτές, υποκαταστήματα (τα οποία πρέπει να εφοδιάζει με προϊόντα) αλλά και με παραγγελίες προς τους προμηθευτές κ.λπ.

Η υλοποίηση του πληροφοριακού συστήματος θα γίνει με χρήση της Java και ιδιαίτερα της τεχνολογίας Servlet, η οποία παρέχει τη δυνατότητα για δημιουργία δυναμικών σελίδων οι οποίες εξαρτώνται από το αίτημα του χρήστη καθώς και από τα δεδομένα που είναι αποθηκευμένα στη Βάση Δεδομένων. Για Βάση Δεδομένων θα χρησιμοποιήσουμε τη MySQL.

SUMMARY

The goal of this present thesis is to implement an information system for car parts (storage unit) by the usage of Java and Servlets. The information system has the ability to store information regarding products, suppliers, branches (that need to be supplied with products) and other orders towards the suppliers, etc.

The information system is implemented with the use of Java and specifically with the Servlet technology, which provides the ability to create dynamic sites that depend on the user’s request as well as the data that is stored in the Data Base. Regarding the Data Base, MySQL is used.

ΠΕΡΙΕΧΟΜΕΝΑ

1ο Κεφάλαιο Γλώσσα Προγραμματισμού Java & Servlet

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε (εν συντομία) τη γλώσσα προγραμματισμού Java και θα περιγράψουμε περισσότερο τα servlet. Θα ξεκινήσουμε αναφέροντας λίγα λόγια για την ιστορία της και στη συνέχεια θα δούμε τα χαρακτηριστικά τους.

1.2 Ιστορία

Το 1991 η Sun Microsystems έκανε μια έρευνα για το μέλλον των υπολογιστικών εργασιών. Το αποτέλεσμά της ήταν ότι το επόμενο που θα συνέβαινε ήταν οι ευφυείς συσκευές. Για να καλυφθεί η συγκεκριμένη ανάγκη δημιουργήθηκε μια μικρή ομάδα με όνομα «Green Project» με σκοπό τη δημιουργία ενός προτύπου για μια συσκευή καταναλωτή, την οποία θα προσπαθούσαν να εισάγουν στην αγορά.

Μετά από 18 μήνες εργασίας η ομάδα παρουσίασε μια συσκευή που ονομαζόταν Star7 (σαν τα PDA αλλά αρκετά μεγαλύτερη) η οποία εμφάνιζε μια φιγούρα με το όνομα Duke που αντιδρούσε στις προτροπές του χρήστη. Η φιγούρα αυτή σήμερα έχει γίνει μασκότ της γλώσσας προγραμματισμού Java.

Εικόνα 1: Ο Duke η μασκότ της Java (Oracle, 2014)

Η συσκευή αυτή βασιζόταν σε μια νέα γλώσσα προγραμματισμού η οποία είχε δημιουργηθεί από τον James Gosling και ονομάστηκε αρχικά Greentalk με επέκταση αρχείου gt. Στη συνέχεια μετονομάστηκε σε Oak, επειδή υπήρχε μια βελανιδιά έξω από το παράθυρο του γραφείου του.

Η συγκεκριμένη γλώσσα διατηρούσε μεγάλη συγγένεια με τη C++ (την οποία είχε αρχικά δοκιμάσει ο Gosling για γλώσσα προγραμματισμού μικροσυσκευών, αλλά την απέρριψε ως ακατάλληλη). Είχε όμως πιο έντονο αντικειμενοστρεφή χαρακτήρα και χαρακτηριζόταν από την απλότητά της. Όταν λίγο αργότερα η ομάδα ανάπτυξης της Oak ενημερώθηκε πως το συγκεκριμένο όνομα είναι ήδη κατοχυρωμένο αναγκάστηκε να ψάξει για ένα νέο όνομα. Οι προτάσεις περιλάμβαναν τα ονόματα: "dynamic", "revolutionary", "Silk", "jolt", "DNA". Από αυτά δύο επιλέχθηκαν το Java και το Silk. Επειδή το Java ήταν μοναδικό, προτιμήθηκε. Java είναι ένα νησί στην Ινδονησία στο οποίο έγινε η πρώτη παραγωγή καφέ (JavaTPoint, n.a).

Εικόνα 2: Java 8 (Oracle, n.a)

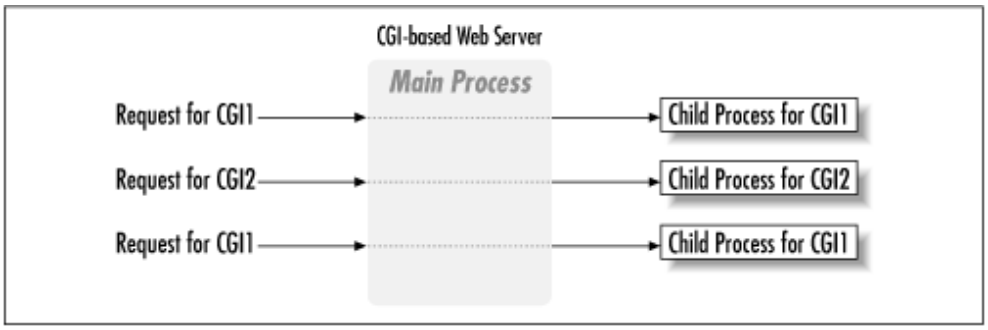
Μέχρι το 1993 η Java δεν είχε την αποδοχή που όλοι περίμεναν. Όμως με την έκρηξη του World Wide Web και την κατανόηση από την Sun ότι το προϊόν που είχαν σχεδιάσει μπορούσε να χρησιμοποιηθεί στο νέο περιβάλλον για να παρέχει δυναμικό περιεχόμενο, επιτάχυνε τις εξελίξεις.

Το 1996 η Sun παρουσιάζει, επίσημα την πρώτη έκδοση της Java. Η έκδοση αυτή θα εμπλουτιστεί με νέες προσθήκες (ενσωμάτωση Swing graphical API κ.λ.π) και το 1998 θα παρουσιαστεί η δεύτερη έκδοση. Στη συνέχεια θα ακολουθήσουν πολλές νέες εκδόσεις και update στις εκδόσεις αυτές. Σήμερα είμαστε στην έκδοση 8 και στο update 40 της συγκεκριμένης έκδοσης (Greanier, 2005)

Ολοκληρώνοντας την ιστορία της Java, θα ξεκινήσουμε την παρουσίαση της ιστορικής διαδρομής των servlet καθώς και την ανάγκη που οδήγησε στη δημιουργία τους.

Ο αρχικός σκοπός του διαδικτύου ήταν στο να παρέχει πρόσβαση και δυνατότητα μεταφοράς/αντιγραφής αρχείων από ένα υπολογιστή σε ένα άλλο. Με τη χρήση του πρωτοκόλλου HTTP και με τη βοήθεια της HTML, οι άνθρωποι μπορούσαν να αποκτήσουν πρόσβαση και να δουν/περιηγηθούν σε αρχεία και περιεχόμενο web server. Όμως η συγκεκριμένη χρήση, άρχισε να γεννά την ανάγκη για δεδομένα πραγματικού χρόνου.

Τη λύση στο πρόβλημα αυτό ήρθε να δώσει το Common Gateway Interface (CGI) μέσω του οποίου μπορούσαμε να δημιουργήσουμε σελίδες με δυναμικό περιεχόμενο. Με το CGI ο web server περνά συγκεκριμένα αιτήματα σε ένα εξωτερικό πρόγραμμα. Η έξοδος του προγράμματος αυτού αποστέλλεται, σε μορφή στατικού αρχείου στον πελάτη που το ζήτησε. Στο σημείο αυτό θα πρέπει να αναφέρουμε ότι η συγκεκριμένη χρησιμότητα του CGI προέκυψε κατά λάθος, ο αρχικός σχεδιασμός είχε σα σκοπό τον καθορισμό μιας τυποποιημένης μεθόδου επικοινωνίας web server με εξωτερικές εφαρμογές. Ο λόγος αυτός, εξηγεί και τον όχι τόσο καλό κύκλο ζωής τους. Όταν ο web server, δέχεται ένα αίτημα για πρόσβαση σε μια εφαρμογή CGI, πρέπει να δημιουργήσει μια νέα διαδικασία για να εκτελέσει το CGI και να του περάσει μέσω τιμών περιβάλλοντος όλες τις πληροφορίες που είναι απαραίτητες. Η δημιουργία νέων διαδικασιών για κάθε αίτημα απαιτεί χρόνο και πόρους, τα οποία περιορίζουν τον ταυτόχρονο αριθμό χρηστών που μπορεί να εξυπηρετήσει ένας server.



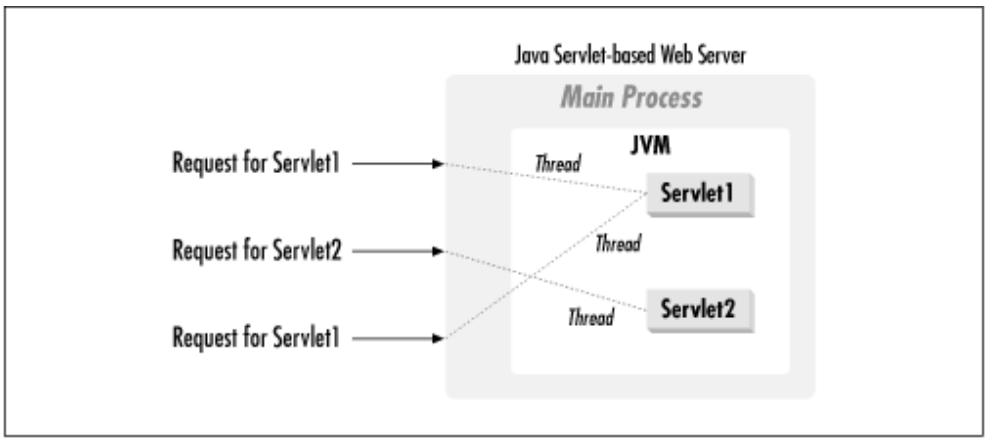
Εικόνα 3: Κύκλος ζωής του CGI (Hunter & Crawford, 1998)

Η πιο συχνά χρησιμοποιούμενη γλώσσα συγγραφής των CGI ήταν η Perl.

Στα επόμενα χρόνια, η Microsoft υλοποίησε μια τεχνική για τη δημιουργία δυναμικού περιεχομένου, την οποία ονόμασε Active Server Pages (ASP). Με

την ASP, μια HTML σελίδα, μπορεί να περιέχει αποσπάσματα κώδικα (συνήθως VBSctipt ή Jscript). Ο κώδικας αυτός διαβάζεται και εκτελείται από τον web server πριν από την αποστολή της σελίδας στον πελάτη.

Ακολούθησαν και άλλες προσπάθειες από διάφορες εταιρείες για υποστήριξη δυναμικού περιεχομένου (όπως το server side JavaScript από τη Netscape). Τον Ιούνιο του 1997 η Sun Microsystems έδωσε την 1η έκδοση των servlet. Ένα servlet είναι μια κλάση Java, που μπορεί να φορτωθεί δυναμικά και να επεκτείνει τη λειτουργικότητα ενός web server. Σε αντίθεση με τα CGI, που απαιτούν πολλές διαδικασίες για να χειριστούν διαφορετικά αιτήματα, τα servlet δημιουργούν thread για την εξυπηρέτηση των αιτημάτων. Επίσης εκτελούμενα μέσα σε ένα Java Virtual Machine μέσα στον web server παρέχουν ασφάλεια και φορητότητα (Hunter & Crawford, 1998).



Εικόνα 4: Κύκλος ζωής του servlet (Hunter & Crawford, 1998)

1.3 Servlet

Στην ενότητα αυτή θα εμβαθύνουμε περισσότερο στα servlet, θα περιγράψουμε την ανάγκη που μας οδήγησε στη δημιουργία τους, τον τρόπο λειτουργίας τους, τα πλεονεκτήματά τους σε σχέση με παρόμοιες τεχνολογίες και τέλος μια σύντομη περιγραφή του τρόπου προγραμματισμού με αυτά.

1.3.1 Γιατί να υπάρχουν δυναμικές ιστοσελίδες

Όπως αναφέραμε, τα servlet είναι προγράμματα τα οποία φτιάχνουν δυναμικά ιστοσελίδες. Γιατί όμως να περιμένουμε να ζητηθεί μια σελίδα για να κατασκευαστεί;

Αρκετές αιτήσεις πελατών μπορούν να εξυπηρετηθούν από προκατασκευασμένες σελίδες. Όμως, σε πολλές περιπτώσεις δεν είναι αρκετό να έχουμε ένα στατικό αποτέλεσμα και πρέπει να παράγουμε μια σελίδα για κάθε αίτηση. Ας δούμε μερικά παραδείγματα:

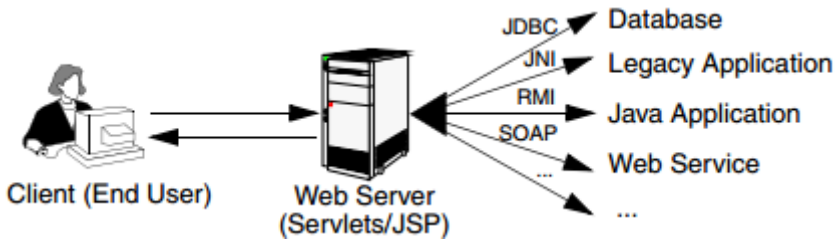
- Η ιστοσελίδα βασίζεται σε δεδομένα που στέλνονται από τον πελάτη. Για παράδειγμα η ιστοσελίδα αποτελεσμάτων των μηχανών αναζήτησης καθώς και των προγραμμάτων web mail, εξαρτώνται από τις συγκεκριμένες αιτήσεις του χρήστη. Δε γνωρίζουμε τι πρέπει να εμφανίσουμε μέχρι να λάβουμε και να διαβάσουμε τα δεδομένα που έχει στείλει ο χρήστης.

- Η ιστοσελίδα παράγεται από δεδομένα που μεταβάλλονται συχνά: Αν η σελίδα αλλάζει με κάθε αίτηση, τότε πρέπει να δομήσουμε την απάντηση κατά το χρόνο της αίτησης.

- Η ιστοσελίδα χρησιμοποιεί πληροφορίες από βάσεις δεδομένων.

1.3.2 Τρόπος λειτουργίας των servlet

Τα servlets και τα προγράμματα σε Java που εκτελούνται σε διακομιστές Ιστού λειτουργούν ως ενδιάμεσο επίπεδο μεταξύ των αιτήσεων που έρχονται από τους φυλλομετρητές Ιστού και των Βάσεων Δεδομένων που υπάρχουν στο διακομιστή HTTP. Η δουλειά τους είναι η εκτέλεση των ακόλουθων εργασιών:



Εικόνα 5: Ο ρόλος του ενδιάμεσου λογισμικού στον Ιστό (Hall & Brown, 2003)

- Ανάγνωση των δεδομένων που έχουν σταλεί από τον πελάτη. Ο χρήστης εισάγει τα δεδομένα σε μια φόρμα HTML η οποία υπάρχει σε κάποια ιστοσελίδα. Τα δεδομένα αυτά όμως θα μπορούσαν να προέρχονται από ένα applet ή από κάποιο άλλο πρόγραμμα πελάτη HTTP.

- Ανάγνωση των έμμεσων δεδομένων αιτήσεων HTTP που έχουν σταλεί από το φυλλομετρητή. Ας δούμε όμως σε τι αναφερόμαστε όταν λέμε έμμεσα δεδομένα. Όταν ο πελάτης στέλνει ένα αίτημα, το αίτημα αυτό περιλαμβάνει δύο ειδών δεδομένα, τα δεδομένα που έχει εισάγει ο χρήστης και τις πληροφορίες HTTP που υπάρχουν στο

παρασκήνιο. Στις HTTP πληροφορίες περιλαμβάνονται τα cookies, οι πληροφορίες για τους τύπους των μέσων και των μεθόδων συμπίεσης κ.λπ.

- Παραγωγή αποτελεσμάτων: Κατά τη διαδικασία αυτή μπορεί να γίνει επικοινωνία με τη βάση δεδομένων, ή την ενεργοποίηση κάποιας υπηρεσίας Ιστού κ.ο.κ. Όμως η βάση δεδομένων δεν καταλαβαίνει το πρωτόκολλο HTTP, οπότε ο φυλλομετρητής Ιστού δε μπορεί να επικοινωνήσει με τη βάση δεδομένων. Χρειαζόμαστε ένα ενδιάμεσο επίπεδο λογισμικού το οποίο θα εξάγει τα εισερχόμενα δεδομένα από το HTTP, θα επικοινωνεί με την εφαρμογή και θα ενσωματώνει τα αποτελέσματα σε κάποιο έγγραφο.

- Αποστολή των ρητών δεδομένων στον πελάτη: Το έγγραφο αυτό μπορεί να σταλεί σε μια ποικιλία μορφών, με συνηθέστερη την HTML. Το αποτέλεσμα των μικροϋπηρεσιών και των σελίδων JSP είναι η ενθυλάκωση των αποτελεσμάτων σε μορφή HTML.

- Αποστολή έμμεσων δεδομένων απάντησης HTTP: Ο web server εκτός από τα πραγματικά δεδομένα στέλνει στον πελάτη και πληροφορίες HTTP, όπως την ενημέρωση του φυλλομετρητή με τον τύπο του εγγράφου, cookies κ.λπ.

1.3.3 Πλεονεκτήματα servlet

Ποια είναι όμως τα πλεονεκτήματα της χρήσης servlet σε σχέση με τις άλλες τεχνολογίες;

- Φορητότητα: Επειδή τα servlet είναι γραμμένα σε Java και συμμορφώνονται με ένα καλά καθορισμένο και αποδεκτό API, είναι φορητά ανάμεσα σε διαφορετικά λειτουργικά συστήματα και εφαρμογές διακομιστή. Με τα servlet μπορούμε να έχουμε το "write once, serve everywhere"

- Ισχύ: Τα servlet μπορούν να αξιοποιήσουν πλήρως την ισχύ του πυρήνα των Java API: δικτύωση και πρόσβαση URL, multithreading, επεξεργασία εικόνας, συμπίεση δεδομένων, σύνδεση με βάση δεδομένων, remote method invocation (RMI) κ.λπ. Εκτός αυτού μπορούμε να επιλέξουμε από μια πληθώρα third-party Java κλάσεων

- Αποδοτικότητα: Όταν φορτωθεί ένα servlet παραμένει στη μνήμη του server σαν ένα αντικείμενο. Στη συνέχεια ο server μπορεί να καλέσει το servlet για να εξυπηρετήσει ένα αίτημα με μια απλή κλήση μεθόδου και όχι με την πολύπλοκη και αρκετά βαριά κλήση των CGI. Πολλαπλές, ταυτόχρονες αιτήσεις, εξυπηρετούνται από ξεχωριστά thread.

- Ασφάλεια: Τα servlet υποστηρίζουν πρακτικές ασφαλούς προγραμματισμού σε διάφορα επίπεδα. Επειδή είναι γραμμένα σε Java τα servlet έχουν κληρονομήσει την ισχυρή ασφάλειά της. Θα πρέπει, να αναφερθούμε και στη δυνατότητα που κληρονομείται από τη Java για το χειρισμό εξαιρέσεων. Αν συμβεί κάποιο λάθος, αυτό μπορεί να αλιευθεί με ασφάλεια (μέσω κάποιας εξαίρεσης) και να εμφανιστεί ένα φιλικό προς το χρήστη μήνυμα λάθους.

- Επεκτασιμότητα και Ευελιξία: Το Servlet API έχει σχεδιαστεί για να είναι εύκολα επεκτάσιμο. Περιλαμβάνει κλάσεις που έχουν βελτιστοποιηθεί για servlets HTTP. Υπάρχει όμως η δυνατότητα για να βελτιστοποιηθεί (από τη SUN ή κάποιον άλλο) για κάποιο άλλο τύπο (Hall & Brown, 2003).

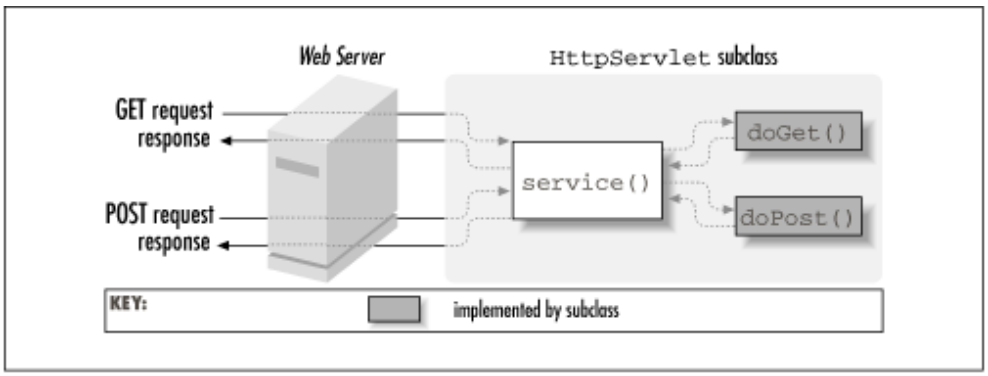
1.3.4 Παράδειγμα Χρήσης

Τα servlet χρησιμοποιούν κλάσεις και interfaces από δύο packages:

- javax.servlet: περιέχει κλάσεις για την υποστήριξη γενικών και ανεξάρτητων πρωτοκόλλου servlet.

- javax.servlet.http: οι κλάσεις του package αυτού έχουν κάνει extend τις κλάσεις του προηγούμενου package για να προσθέσουν HTTP λειτουργικότητα.

Σε αντίθεση με τα παραδοσιακά προγράμματα σε Java τα servlet δεν έχουν main μέθοδο. Ο server είναι αυτός που κατά την επεξεργασία των αιτημάτων καλεί τη μέθοδο service() του servlet και μέσω αυτής τη μέθοδο doGet() για την εξυπηρέτηση Get αιτημάτων και τη μέθοδο doPost για την εξυπηρέτηση Post αιτημάτων.



Εικόνα 6: Ένα HTTP servlet εξυπηρετεί μια αίτηση
(Hunter & Crawford, 1998)

Ένα servlet πρέπει να κάνει override τις δύο μεθόδους (ή μια από τις δύο ανάλογα με ποια αιτήματα θα εξυπηρετεί. για να εξυπηρετήσει τα αιτήματα με τον τρόπο που θέλει. Η μέθοδος service() δέχεται δύο παραμέτρους ένα αντικείμενο request και ένα αντικείμενο response. Το αντικείμενο request

ενημερώνει το servlet σχετικά με το αίτημα, ενώ το αντικείμενο response χρησιμοποιείται για την επιστροφή της απάντησης.

Ο βασικός τύπος ενός HTTP servlet παράγει μια πλήρη σελίδα HTML.

Ας δούμε πώς συντάσσεται ένα servlet:

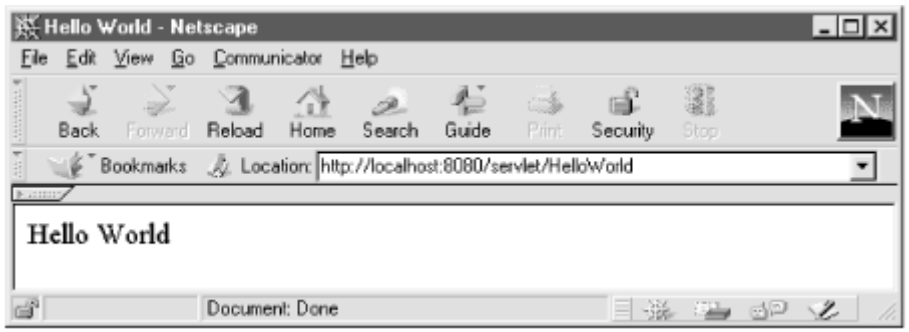
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

  public void doGet(HttpServletRequest req,
HttpServletResponse res)throws ServletException, IOException
{

    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    out.println("<HTML>");
    out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("<BIG>Hello World</BIG>");
    out.println("</BODY></HTML>");
  }
}
```

Όπως βλέπουμε από τον κώδικα παραπάνω ένα HTTP servlet κάνει extend την HTTPServlet κλάση και κάνει override τη μέθοδο doGet(). Κάθε φορά που ο server δέχεται ένα αίτημα Get για το servlet, καλεί τη μέθοδο doGet() περνώντας σαν παραμέτρους τα αντικείμενα HttpServletRequest και HttpServletResponse.

Το αντικείμενο HttpServletRequest περιέχει το αίτημα του χρήστη. Δίνει τη δυνατότητα στο servlet να έχει πρόσβαση σε όλα τα στοιχεία, τις παραμέτρους τα δεδομένα που έχει στείλει ο χρήστης και τα συνοδεύουν. Στο παράδειγμά μας δεν το χρησιμοποιούμε γιατί θέλουμε να του εμφανίσουμε μια σελίδα που να γράφει Hello World ανεξάρτητα από το αίτημα που έχει κάνει.

Το αντικείμενο HttpServletResponse απεικονίζει την απάντηση του servlet. Μέσω αυτού το servlet στέλνει δεδομένα στον πελάτη. Στο παράδειγμά μας, το servlet καλεί τη μέθοδο setContentType() για να καθορίσει τον τύπο του εγγράφου επιστροφής ότι θα είναι "text/html". Μετά καλεί την getWriter() για να πάρει ένα αντικείμενο τύπου PrintWriter, που μοιάζει με το PrintStream και μετατρέπει τους Unicode χαρακτήρες στην τοπική κωδικοποίηση. Στη συνέχεια σχηματίζει την html σελίδα.

Για να μπορέσουμε να το εκτελέσουμε και να δούμε το αποτέλεσμα στην οθόνη μας θα πρέπει να υπάρχει εγκατεστημένος στον υπολογιστή μας ένας web server. Μια πιο απλή λύση είναι να κατεβάσουμε το IDE Netbeans που περιέχει έναν web server (GlassFish) και μας δίνει δυνατότητα εκτέλεσης του servlet μέσα από το περιβάλλον του (αναλαμβάνει το compile και το deploy των servlet στον web server).



Εικόνα 7: Αποτέλεσμα Παραδείγματος (Hunter & Crawford, 1998)

Σαν τελευταίο παράδειγμα προγραμματισμού, ας δούμε πώς χειρίζεται ένα servlet τα δεδομένα που δέχεται από μια φόρμα.

Έστω ότι έχουμε την ακόλουθη Html φόρμα:

```
<HTML>
<HEAD>
<TITLE>Introductions</TITLE>
</HEAD>
```

```
<BODY>
<FORM METHOD=GET ACTION="/servlet/Hello">
If you don't mind me asking, what is your name?
<INPUT TYPE=TEXT NAME="name"><P>
<INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

Όταν ο χρήστης πατά το submit button τότε καλείται το servlet Hello μέσω της μεθόδου GET. Ας δούμε τώρα πώς το servlet μπορεί να δεχτεί τα δεδομένα που του έστειλε η φόρμα, δηλαδή το text box name.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Hello extends HttpServlet {

  public void doGet(HttpServletRequest req,
HttpServletResponse res)
   throws ServletException, IOException {

    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    String name = req.getParameter("name");
    out.println("<HTML>");
    out.println("<HEAD><TITLE>Hello, " + name +
"</TITLE></HEAD>");
    out.println("<BODY>");
    out.println("Hello, " + name);
    out.println("</BODY></HTML>");
  }

  public String getServletInfo() {
    return "A servlet that knows the name of the person to
whom it's" +
           "saying hello";
  }
}
```

Όπως βλέπουμε, σε μια μεταβλητή τύπου String με όνομα name δίνουμε την τιμή που μας έχει δώσει ο χρήστης μέσω της φόρμας με την κλήση της req.getParameter("name").

Όπως έχουμε πει το req μας δίνει πρόσβαση στο αντικείμενο που περιέχει ότι μας έχει στείλει ο χρήστης. Στη συνέχεια η getParameter είναι μια μέθοδος μέσω της οποίας μπορούμε να έχουμε πρόσβαση στις παραμέτρους που περιέχουν τα δεδομένα του χρήστη, δηλαδή στα στοιχεία της φόρμας. Η μέθοδος αυτή, επιστρέφει τα δεδομένα που έχουν συμπληρωθεί ή NULL.

Το αποτέλεσμα θα είναι να εμφανιστεί στην οθόνη του χρήστη, το όνομα που έχει συμπληρώσει (Hunter & Crawford, 1998).

2ο Κεφάλαιο Βάση δεδομένων mysql

2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα κάνουμε μια σύντομη ιστορική αναδρομή στα Συστήματα Διαχείρισης Βάσεων Δεδομένων και στη MySQL που θα είναι η βάση δεδομένων που θα χρησιμοποιήσουμε.

2.2 Ιστορική Αναδρομή ΣΔΒΔ

Με την εμφάνιση των υπολογιστικών συστημάτων, αποκαλύφθηκε το πόσο σημαντικό ήταν ο ρόλος της αποθήκευσης και της επεξεργασίας των δεδομένων. Το πρώτο σύστημα Διαχείρισης Βάσεων Δεδομένων κατασκευάστηκε από τον Charles Bachman της εταιρείας General Electric στις αρχές του 1960 και ονομάστηκε Integrated Data Store (IDS). Το IDS βασιζόταν στο δικτυωτό μοντέλο δεδομένων το οποίο είχε εισαχθεί από το Συνέδριο Γλωσσών των Συστημάτων Δεδομένων και επηρέασε πολλά Σ.Δ.Β.Δ που αναπτύχθηκαν στη δεκαετία του 1960. Στο συγκεκριμένο μοντέλο οι σχέσεις των δεδομένων απεικονίζονται με ένα γράφο.

Στα τέλη της δεκαετίας του 1960 η IBM ανέπτυξε το σύστημα Information Management System (IMS). Το σύστημα αυτό υποστηρίζει ακόμα και σήμερα κάποιες εφαρμογές (όπως το σύστημα κράτησης της American Airlines, SABRE). Το Information Management System της IBM βασίζεται στο Ιεραρχικό μοντέλο δεδομένων.

Τα προβλήματα που παρουσιάστηκαν με τα πρώτα συστήματα καθώς και τα αντίστοιχα μοντέλα ήταν:

- Ο χρήστης έπρεπε να γνωρίζει τη φυσική δομή της βάσης δεδομένων για να μπορέσει να αποστείλει τα ερωτήματα.

- Η προσθήκη ενός επιπλέον πεδίου απαιτούσε την επαναδημιουργία του συστήματος πρόσβασης.

- Ο τρόπος αποθήκευσης είχε άμεση εξάρτηση από τον τύπο των δεδομένων.

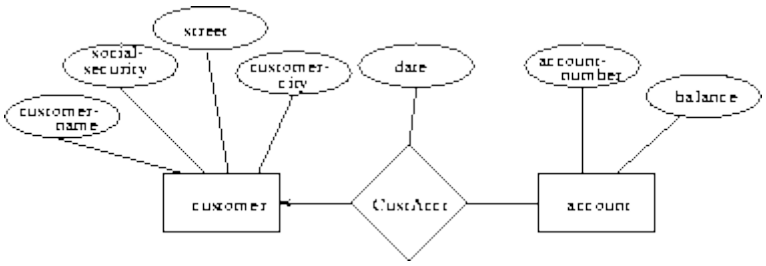
Τα πρώτα συστήματα διαχείρισης βάσεων δεδομένων ήταν πολύ δύσκολα και δύσχρηστα και ιδιαίτερα όταν χρειαζόταν η τροποποίηση των δεδομένων ή η προσθήκη νέων εφαρμογών.

Το 1970 ο Edgar Codd από τα εργαστήρια San Jose της IBM πρότεινε ένα νέο σχήμα αναπαράστασης των δεδομένων το οποίο ονόμασε σχεσιακό μοντέλο. Η παρουσίαση του συγκεκριμένου μοντέλου προκάλεσε την ταχεία ανάπτυξη του σχεσιακού τύπου Σ.Δ.Β.Δ. Για το λόγο αυτό το 1981 ο Codd τιμήθηκε με το βραβείο Turing.

Βασιζόμενα στο σχήμα που πρότεινε ο Codd κατασκευάστηκαν δύο νέα συστήματα σχεσιακών βάσεων δεδομένων. Το ένα ήταν η INGRES από την UBC και το άλλο το SYSTEM R από την IBM. Η INGRES χρησιμοποιούσε τη γλώσσα ερωτημάτων QUEL και οδήγησε στη δημιουργία συστημάτων όπως MS-SQL Server, Sybase κ.λ.π.

Το SYSTEM R χρησιμοποιούσε τη SEQUEL(Structured English Query Language) σαν γλώσσα ερωτημάτων, η οποία αργότερα μετονομάστηκε σε SQL. Το SYSTEM R οδήγησε στη δημιουργία των ακόλουθων συστημάτων DB2, Oracle κ.λ.π.

Το 1976 ο Peter Chen πρότεινε έναν νέο μοντέλο για το σχεδιασμό βάσεων δεδομένων, το μοντέλο Entity Relationship. Το μοντέλο αυτό επιτρέπει στο σχεδιαστή της βάσης δεδομένων να επικεντρωθεί στη χρήση των δεδομένων και όχι στη λογική δομή των πινάκων.



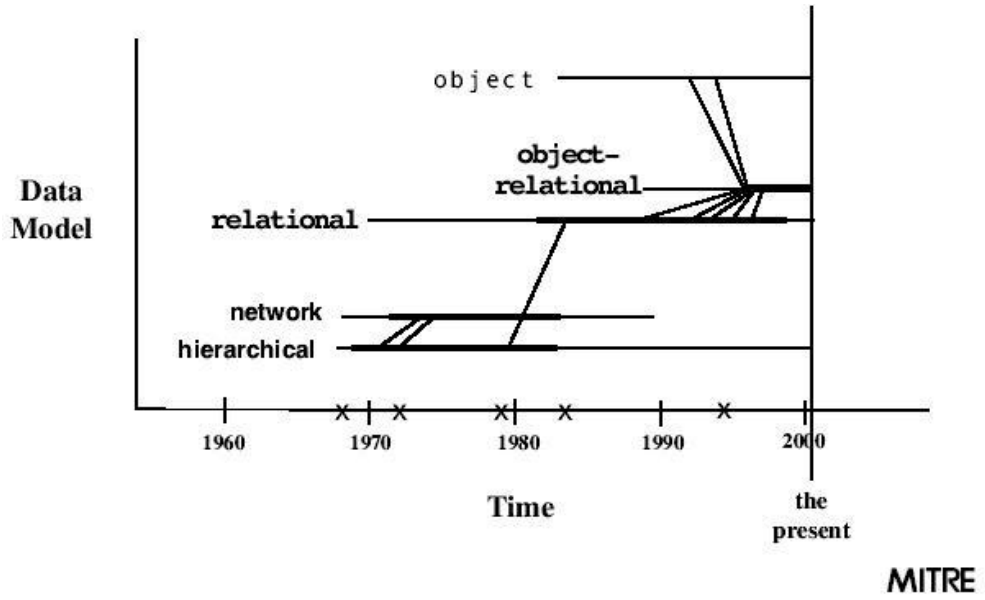
Εικόνα 8: Entity Relationship Model (Ostema, 2004)

Στη δεκαετία του 1980 υπήρξε μια έκρηξη της χρήσης υπολογιστών και κατ' επέκταση μια αύξηση της χρήσης βάσεων δεδομένων. Στη δεκαετία αυτή κυριάρχησε το σχεσιακό μοντέλο. Επίσης υπήρξε η τυποποίηση της SQL από την ANSI και τον ISO. Προς τα τέλη της δεκαετίας, έγινε φανερό ότι υπήρχαν ορισμένοι τομείς (όπως ιατρική, πολυμέσα, φυσική) όπου οι σχεσιακές βάσεις δεδομένων δεν ήταν πρακτικές λόγω των ίδιων των δεδομένων. Αυτό οδήγησε σε έρευνα για αναζήτηση ενός νέου τύπου βάσεων δεδομένων. Η έρευνα αυτή κατέληξε στην πρόταση των Αντικειμενοστραφών Βάσεων Δεδομένων, στις οποίες οι χρήστες θα μπορούσαν να καθορίσουν τις δικές τους μεθόδους πρόσβασης σε αυτά, τους τρόπους απεικόνισης και χρήσης τους. Η παρουσίαση αυτή συν έπεσε με την εισαγωγή του αντικειμενοστραφούς προγραμματισμού.

Το 1990 παρουσιάστηκαν οι πρώτες αντικειμενοστραφείς βάσεις δεδομένων. Την ίδια χρονική περίοδο λίγες είναι οι εταιρείες που προσφέρουν Σ.Δ.Β.Δ που καταφέρνουν να ξεπεράσουν τα προβλήματα και να επιζήσουν. Όσες τα καταφέρουν προσφέρουν πιο πολύπλοκα προϊόντα σε ακριβότερες τιμές. Για την κάλυψη των αναγκών χρηστών μικρότερων επιχειρήσεων προσφέρονται νέα εργαλεία όπως η Access από τη Microsoft.

Στις επόμενες δεκαετίας με την έκρηξη του διαδικτύου παρουσιάζονται νέα προϊόντα τα οποία δίνουν έμφαση (κυρίως) στον τρόπο διασύνδεσης της

τελικής εφαρμογής με τη Βάση Δεδομένων. Επίσης εμφανίζονται Open Source προϊόντα όπως η MySQL (Ostema, 2004)



Εικόνα 9: Data Model vs Time (Ostema, 2004)

2.3 Ιστορική Αναδρομή της MySQL

Η MySQL δημιουργήθηκε από μια Σουηδική εταιρία, τη MySQL AB το 1995. Οι προγραμματιστές που τη δημιούργησαν ήταν δύο Σουηδοί και ένας Φινλανδός, οι Michael Widenius (Μόντυ), David Axmark και Allan Larsson. Ο κυριότερος σκοπός ήταν να παρέχει αποτελεσματικές και αξιόπιστες επιλογές στη διαχείριση δεδομένων για οικιακή και επαγγελματική χρήση.

Το όνομά της προέρχεται από το όνομα της κόρης του Μόντυ My. Ο οποίος δημιούργησε και άλλες βάσεις δεδομένων δίνοντας ονόματα των άλλων παιδιών του MariaDB και MaxDB (Dybka, 2014).

Εικόνα 10: MySQL (Dybka, 2014)

Πολλές αρχικές εκδόσεις (alpha και beta) της πλατφόρμας κυκλοφόρησαν κατά το έτος 2000. Αυτές οι εκδόσεις ήταν συμβατές με σχεδόν όλες τις διαδεδομένες πλατφόρμες εκείνης της εποχής.

Η απόφαση να γίνει ανοικτού κώδικα, ακολουθώντας τους όρους του GPL το 2000, οδήγησε σε σημαντική πτώση των εσόδων της εταιρίας. ωστόσο κατάφερε να ανακτήσει το χαμένο έδαφος. Ο χαρακτήρας αυτός, όμως τη βοήθησε γιατί δέχτηκε τη συνεισφορά από τρίτους προγραμματιστές, βελτιώνοντας πολλά χαρακτηριστικά της.

Η MySQL απέκτησε σταθερή δημοτικότητα τόσο στους οικιακούς χρήστες, όσο και σε επαγγελματίες και το έτος 2001, η πλατφόρμα είχε φτάσει αισίως τα 2 εκατομμύρια ενεργές εγκαταστάσεις. Το 2002, η εταιρεία άνοιξε γραφεία στις Η.Π.Α., ανακοινώνοντας ότι τα τα ενεργά μέλη της πλατφόρμας ξεπέρασαν τα 3 εκατομμύρια χρήστες με έσοδα που ανέρχονται σε $6,5 εκατομμύρια.

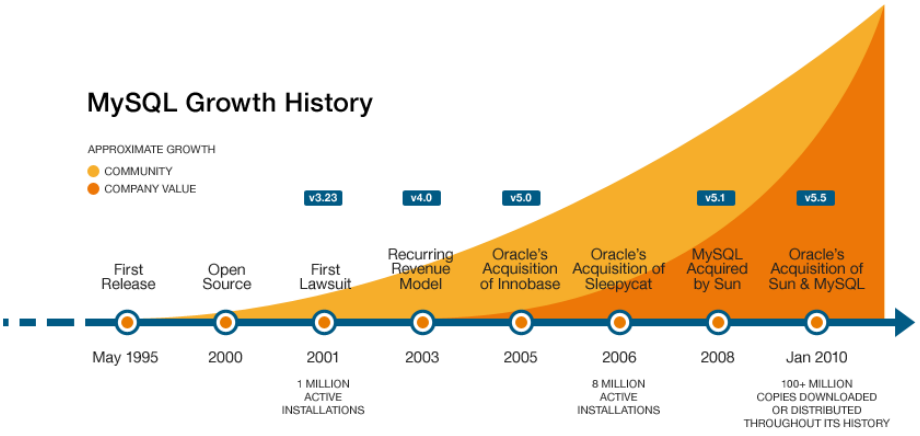
Η πλατφόρμα συνέχισε να κερδίζει δημοτικότητα και μέχρι το τέλος του 2003, είχαν φτάσει τα έσοδά της σε 12 εκατομμύρια δολάρια με 4 εκατομμύρια ενεργές εγκαταστάσεις. Το 2004 η εταιρεία αποφάσισε να επικεντρωθεί περισσότερο στο μοντέλο επαναλαμβανόμενων χρεώσεων

στους χρήστες, αντί για χρέωση μόνο στην αρχική αδειοδότηση, έτσι το έτος έκλεισε με καθαρά έσοδα ύψους $20 εκατομμυρίων.

Το 2005, η Oracle αγόρασε την Innobase, την εταιρεία που διαχειρίζεται την MySQL Innobase storage backend, τη μηχανή αποθήκευσης η οποία προσέφερε πολλές σημαντικές λειτουργίες στη MySQL (transactions και foreign keys). Το ίδιο έτος, κυκλοφόρησε η MySQL 5, μία έκδοση με νέα χαρακτηριστικά (ιδιαίτερα για εταιρικούς χρήστες). Το επόμενο έτος, η σύμβαση μεταξύ MySQL και Innobase ανανεώθηκε.

Κατά το έτος 2006, η δημοτικότητα της εταιρείας εξακολούθησε να αυξάνεται με 8 εκατομμύρια ενεργές εγκαταστάσεις το 2006.

Τον Ιανουάριο του 2008, η MySQL εξαγοράστηκε από την Sun Microsystems για 1 δις δολάρια. Ωστόσο, η απόφαση επικρίθηκε από τους Michael Widenius και David Axmark, οι συνιδρυτές της MySQL AB.



Εικόνα 11: Ιστορικό Ανάπτυξης της MySQL (Database Friends, 2014)

Όμως, η εξαγορά της MySQL από την Sun δεν αποδείχθηκε πολύ καρποφόρα και τον Απρίλιο του 2009 επετεύχθη νέα συμφωνία μεταξύ της Sun

Microsystems και την Oracle Corporation, σύμφωνα με την οποία η Oracle θα εξαγόραζε τη Sun Microsystems μαζί με την MySQL.

Ο Widenius έφυγε από τη Sun όταν εξαγοράστηκε από την Oracle και ανέπτυξε μια νέα βάση δεδομένων την MariaDB. Η MariaDB θεωρείται ένα fork της MySQL.

Οι παλαιότερες εκδόσεις του MySQL αναπτύχθηκαν μόνο για τα συμβατικά μηχανήματα. Ωστόσο, με την έλευση του Cloud Computing, η MySQL έγινε και αυτή συμβατή με τις διάφορες υπηρεσίες cloud computing, όπως η Amazon EC^2. Διάφορα μοντέλα ανάπτυξης έχουν χρησιμοποιηθεί για την εφαρμογή της MySQL σε πλατφόρμες cloud computing. Ίσως το πιο δημοφιλές από αυτά, είναι το «Virtual Machine Image», το οποίο είναι ένα image μιας μηχανής που περιέχει τη MySQL προ-εγκατεστημένη (DataBase Friends, 2014).

3ο Κεφάλαιο Σχεδιασμός και υλοποίηση εφαρμογής

3.1 Εισαγωγή

Στο κεφάλαιο αυτό, θα παρουσιάσουμε τα στάδια του σχεδιασμού και της υλοποίησης της εφαρμογής. Αρχικά θα παρουσιάσουμε τον τρόπο σχεδιασμού και υλοποίησης της βάσης δεδομένων. Στη συνέχεια θα μελετήσουμε τον σχεδιασμό και τον τρόπο υλοποίησης του προγράμματος. Τέλος θα κλείσουμε με μια σύντομη παρουσίαση του τρόπου λειτουργίας της εφαρμογής.

3.2 Σχεδιασμός Βάσης Δεδομένων

Για την υλοποίηση της εφαρμογής της αποθήκης ανταλλακτικών, είναι απαραίτητο να αποθηκεύονται οι ακόλουθες πληροφορίες στη βάση δεδομένων:

- Τα στοιχεία κάθε προϊόντος, τα οποία περιλαμβάνουν Όνομα, είδος, κατασκευαστής/παραγωγός και τιμή.
- Τα καταστήματα που προμηθεύει η αποθήκη , Όνομα., διεύθυνση
- Τους προμηθευτές, Ονοματεπώνυμο, διεύθυνση και τηλέφωνο
- Τις παραγγελίες προς τους προμηθευτές τις αποθήκης: κωδικό του προϊόντος και ποσότητα.

- Τους εφοδιασμούς προς τα καταστήματα. Κάθε εφοδιασμός πρέπει να έχει μοναδικό κωδικό και ημερομηνία.

Επίσης θέλουμε να αποθηκεύουμε:

- Κάθε προμηθευτής συνδέεται με πολλά προϊόντα (τουλάχιστον 1) και κάθε προϊόν παρέχεται από πολλούς προμηθευτές (τουλάχιστον 1). Κάθε προμηθευτής έχει μια τιμή πώλησης η οποία αλλάζει συχνά.
- Μια παραγγελία γίνεται προς ένα και μόνο ένα προμηθευτή.
- Ένας εφοδιασμός αναφέρεται προς ένα και μόνο ένα υποκατάστημα.

Λαμβάνοντας υπόψη τις παραπάνω απαιτήσεις προχωρήσαμε στο σχεδιασμό της βάσης δεδομένων warehouse, η οποία περιλαμβάνει τους παρακάτω πίνακες.

3.2.1 Product

Στον πίνακα αυτό θα κρατάμε πληροφορίες για τα προϊόντα. Περιλαμβάνει τα παρακάτω πεδία:

Πεδίο	**Τύπος**	**Περιγραφή**
productId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός προϊόντος
name	varchar(40) utf8_bin	Όνομα προϊόντος
type	varchar(40) utf8_bin	Τύπος προϊόντος
manufacturer	varchar(40) utf8_bin	Κατασκευαστής

		προϊόντος
price	double	Τιμή
valid	tinyint(1)	Έγκυρο/άκυρο

Το πεδίο valid μπορεί να πάρει τιμές true/false και χρησιμοποιείται όταν θέλουμε να διαγράψουμε ένα προϊόν. Με τον τρόπο αυτό, όταν διαγράφουμε ένα προϊόν, θέτουμε το πεδίο valid σε τιμή false, έτσι η πληροφορία παραμένει στη βάση δεδομένων μας, απλά είναι ανενεργή.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**productId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**name**	varchar(40)	utf8_bin		Όχι	*Καμία*	
3	**type**	varchar(40)	utf8_bin		Όχι	*Καμία*	
4	**manufacturer**	varchar(40)	utf8_bin		Όχι	*Καμία*	
5	**price**	double			Όχι	*Καμία*	
6	**valid**	tinyint(1)			Όχι	1	

3.2.2 shop

Στον πίνακα αυτό αποθηκεύουμε πληροφορίες για τα καταστήματα που προμηθεύει η αποθήκη. Περιλαμβάνει τα ακόλουθα πεδία:

Πεδίο	**Τύπος**	**Περιγραφή**
shopId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός Καταστήματος

name	varchar(40) utf8_bin	Όνομα Καταστήματος
city	varchar(40) utf8_bin	Πόλη
street	varchar(40) utf8_bin	Οδός
number	int(11)	Αριθμός
tk	int(11)	Ταχυδρομικός Κώδικας
valid	tinyint(1)	Έγκυρο/άκυρο

Το πεδίο valid έχει ίδια λογική με αυτή στον πίνακα product. Δηλαδή όταν θέλουμε να διαγράψουμε ένα κατάστημα, τότε απλά το θέτουμε στην τιμή false.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**shopId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**name**	varchar(50)	utf8_bin		Όχι	*Καμία*	
3	**city**	varchar(40)	utf8_bin		Όχι	*Καμία*	
4	**street**	varchar(40)	utf8_bin		Όχι	*Καμία*	
5	**number**	int(11)			Όχι	*Καμία*	
6	**tk**	int(11)			Όχι	*Καμία*	
7	**valid**	tinyint(1)			Όχι	1	

3.2.3 supplier

Ο πίνακας supplier αποθηκεύει τους προμηθευτές. Έχει τα ακόλουθα πεδία.

Πεδίο	**Τύπος**	**Περιγραφή**
supplierId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός Προμηθευτή
name	varchar(40) utf8_bin	Όνομα Προμηθευτή
city	varchar(40) utf8_bin	Πόλη
street	varchar(40) utf8_bin	Οδός
number	int(11)	Αριθμός
tk	int(11)	Ταχυδρομικός Κώδικας
phone	varchar(9) utf8_bin	Τηλέφωνο Προμηθευτή
valid	tinyint(1)	Έγκυρο/άκυρο

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**supplierId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**name**	varchar(60)	utf8_bin		Όχι	*Καμία*	
3	**city**	varchar(40)	utf8_bin		Όχι	*Καμία*	
4	**street**	varchar(40)	utf8_bin		Όχι	*Καμία*	
5	**number**	int(11)			Όχι	*Καμία*	
6	**tk**	int(11)			Όχι	*Καμία*	
7	**phone**	varchar(9)	utf8_bin		Όχι	*Καμία*	
8	**valid**	tinyint(1)			Όχι	1	

3.2.4 supply

Στις απαιτήσεις του συστήματος περιγράψαμε ότι ένα προϊόν προμηθεύεται από διαφορετικούς προμηθευτές και σε διαφορετική τιμή. Για να μπορούμε να διατηρήσουμε τη συγκεκριμένη πληροφορία στη βάση δεδομένων, δημιουργήσαμε τον πίνακα supply όπου αποθηκεύουμε τον κωδικό του προμηθευτή, τον κωδικό του προϊόντος που προμηθεύει και την τιμή του. Ας δούμε τα πεδία που χρησιμοποιούμε:

Πεδίο	**Τύπος**	**Περιγραφή**
supplyId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός Προμήθειας
productId	int(11)	Κωδικός Προϊόντος
supplierId	int(11)	Κωδικός Προμηθευτή
price	double	Τιμή Προϊόντος

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**supplyId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**productId**	int(11)			Όχι	*Καμία*	
3	**supplierId**	int(11)			Όχι	*Καμία*	
4	**price**	double			Όχι	*Καμία*	

3.2.5 shopstocks

Για να μπορούμε να αποθηκεύουμε στη βάση δεδομένων το τρέχον υπόλοιπο των προϊόντων στα καταστήματα, αλλά και στην κεντρική αποθήκη, δημιουργήσαμε τον πίνακα shopstocks. Ο πίνακας αυτός περιέχει και

πληροφορία για τα ελάχιστα stock, που όταν το υπόλοιπο των προϊόντων πέσει κάτω από αυτό, πρέπει να γίνει εφοδιασμός ή παραγγελία.

Ας δούμε τα πεδία που υπάρχουν:

Πεδίο	**Τύπος**	**Περιγραφή**
stockId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός στοκ
shopId	int(11)	Κωδικός Καταστήματος
productId	int(11)	Κωδικός Προϊόντος
stock	int(11)	Τρέχον στοκ
min_stock	int(11)	Ελάχιστο στοκ

Όλα τα καταστήματα έχουν ένα μοναδικό κωδικό, ο οποίος υπάρχει στον πίνακα shop. Στην κεντρική αποθήκη έχουμε αντιστοιχήσει τον κωδικό 0.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**stockId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**shopId**	int(11)			Όχι	*Καμία*	
3	**productId**	int(11)			Όχι	*Καμία*	
4	**stock**	int(11)			Όχι	*Καμία*	
5	**min_stock**	int(11)			Όχι	*Καμία*	

3.2.6 efodiasmos

Ο πίνακας εφοδιασμός περιέχει την ημερομηνία του εφοδιασμού και τον κωδικό καταστήματος, προς το οποίο γίνεται.

Πεδίο	Τύπος	Περιγραφή
eId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός Εφοδιασμού
shopId	int(11)	Κωδικός Καταστήματος
dateOfEfodiasmos	date	Ημερομηνία Εφοδιασμού

Οι πληροφορίες του εφοδιασμού έχουν καταχωριστεί στον ακόλουθα πίνακα.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**eId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**shopId**	int(11)			Όχι	*Καμία*	
3	**dateOfEfodiasmos**	date			Όχι	*Καμία*	

3.2.7 efodiasmosdetails

Όπως αναφέραμε ο συγκεκριμένος πίνακας περιέχει τα προϊόντα τα οποία περιλαμβάνονται σε έναν εφοδιασμό.

Πεδίο	Τύπος	Περιγραφή
edetailsId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός
eId	int(11)	Κωδικός Εφοδιασμού
productId	int(11)	Κωδικός Προϊόντος
quantity	int(11)	Ποσότητα Προϊόντος

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**edetailsId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**eId**	int(11)			Όχι	*Καμία*	
3	**productId**	int(11)			Όχι	*Καμία*	
4	**quantity**	int(11)			Όχι	*Καμία*	

3.2.8 order

Στον πίνακα order αποθηκεύουμε τις παραγγελίες της κεντρικής αποθήκης προς τους προμηθευτές. Για να διατηρήσουμε το μέγεθος της βάσης δεδομένων μικρό και η πληροφορία να αποθηκεύεται σε ένα σημείο, δεν προσθέσαμε στον πίνακα αυτό, τον προμηθευτή και την τιμή του προϊόντος αλλά προσθέσαμε το primary key από τον πίνακα supply. Ο πίνακας αυτός δείχνει ποιος προμηθευτής προμηθεύει ποιο προϊόν και σε ποια τιμή.

Πεδίο	Τύπος	Περιγραφή
orderId	int(11), AUTO_INCREMENT, Primary Key	Μοναδικός κωδικός παραγγελίας
dateOfOrder	date	Ημερομηνία παραγγελίας
supplyId	int(11)	Κωδικός Προμήθειας
quantity	int(11)	Ποσότητα Προϊόντος

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Πρόσθετα
1	**orderId**	int(11)			Όχι	*Καμία*	AUTO_INCREMENT
2	**dateOfOrder**	date			Όχι	*Καμία*	
3	**supplyId**	int(11)			Όχι	*Καμία*	
4	**quantity**	int(11)			Όχι	*Καμία*	

3.3 Σχεδιασμός και Υλοποίηση Εφαρμογής

Μετά από την ολοκλήρωση του σχεδιασμού και την υλοποίηση της βάσης δεδομένων, θα προχωρήσουμε στο σχεδιασμό της εφαρμογής. Η βασική απαίτηση είναι η υλοποίηση της εφαρμογής να γίνει με τη χρήση servlet.

Επειδή η εφαρμογή μας θα περιλαμβάνει και κάποιες σελίδες οι οποίες δε χρειάζονται αλληλεπίδραση με τη βάση δεδομένων αποφασίσαμε να τις υλοποιήσουμε σε jsp, για να προσφέρουν και κάποια δυναμικά χαρακτηριστικά.

3.3.1 Index.jsp

Η αρχική σελίδα της εφαρμογής μας. Εμφανίζει όλες τις διαθέσιμες επιλογές της εφαρμογής σε μορφή υπεσυνδέσμων και θέτει τα session που χρησιμοποιούμε σε κενό.

Λειτουργία	Servlet / jsp
Υποκαταστήματα	
Προσθήκη Υποκαταστήματος	AddShop.jsp
Διαγραφή Υποκαταστήματος	Servlet DeleteShop
Προβολή Υποκαταστήματος	Servlet ShowShop
Ενημέρωση Κινήσεων	Servlet SelectShop
Εφοδιασμός	Servlet SelectShop
Προϊόντα	
Προσθήκη	AddProduct.jsp
Διαγραφή Προϊόντος	Servlet DeleteProduct
Προβολή Προϊόντος	Servlet ShowProduct
Υπόλοιπα Αποθήκης	Servlet ShowMainShop
Προμηθευτές	
Προσθήκη Προμηθευτή	AddSupplier.jsp
Διαγραφή Προμηθευτή	Servlet DeleteSupplier

Προβολή Προμηθευτή	Servlet ShowSupplier

3.3.2 AddShop.jsp

Με το jsp ξεκινάμε τη διαδικασία προσθήκης ενός νέου υποκαταστήματος. Περιέχει μια φόρμα, όπου ο χρήστης συμπληρώνει τα στοιχεία του καταστήματος που θέλει να προσθέσει και όταν πατήσει το submit καλείται το servlet AddShop.

Το servlet θα επιστρέψει το αποτέλεσμα της προσθήκης (θετικό ή αρνητικό στο jsp μέσω session μεταβλητών. Οι μεταβλητές αυτές περιέχουν το μήνυμα λάθους (errorMessage) ή το μήνυμα επιτυχίας (okMessage) καθώς και τις τιμές που έχει συμπληρώσει ο χρήστης. Έτσι στην περίπτωση που έχει γίνει κάποιο λάθος και δε μπορεί να προστεθεί το κατάστημα στη βάση δεδομένων, θα εμφανιστεί το μήνυμα λάθους και η φόρμα με συμπληρωμένα τα στοιχεία που είχαμε δώσει.

3.3.3 AddShop

Το servlet διαβάζει τις τιμές που του στέλνει το AddShop.jsp. Ελέγχει αν έχουμε συμπληρώσει πόλη, αριθμό, οδό και Ταχυδρομικό κώδικα και αν κάποιο δεν έχει συμπληρωθεί:

- Θέτει στις μεταβλητές session τις τιμές που πήρε
- Θέτει στη μεταβλητή session που δείχνει το λάθος, το μήνυμα λάθους

- Επιστρέφει στην AddShop.jsp

Αν όλες οι τιμές έχουν συμπληρωθεί σωστά, τότε δημιουργεί ένα νέο αντικείμενο του τύπου myConnection, και εκτελεί μια εντολή insert στον πίνακα shop, για να προσθέσει το νέο κατάστημα. Κατόπιν με ένα ερώτημα select βρίσκει τον κωδικό που αποδόθηκε στο κατάστημα και θέτει την αντίστοιχη μεταβλητή session.

Αν δεν υπήρξε κάποιο πρόβλημα κατά την εκτέλεση των ερωτημάτων καλούμε το servlet AddProductsToShop για να προσθέσουμε προϊόντα στο κατάστημα που μόλις δημιουργήσαμε

Αν υπήρξε πρόβλημα στην εκτέλεση των ερωτημάτων και δημιουργήθηκε κάποιο SQLException τότε θέτουμε στις μεταβλητές session τις τιμές που μας έδωσε ο χρήστης και στη μεταβλητή με το μήνυμα λάθους το λάθος και επιστρέφουμε στο AddShop.jsp.

3.3.4 AddProductsToShop

Χρησιμοποιείται για να προσθέσει προϊόντα σε ένα υποκατάστημα. Εμφανίζει μια φόρμα, της οποίας το action είναι το ίδιο servlet (το AddProductsToShop).

Τα στοιχεία της φόρμας είναι ένα select το οποίο περιέχει όλα τα προϊόντα που έχουν valid=1. Επίσης η φόρμα περιέχει δύο πεδία τύπου number για να καθορίζουμε το αρχικό στοκ καθώς και το ελάχιστο στοκ για κάθε προϊόν που προσθέτουμε στο κατάστημα.

Στο κάτω μέρος της φόρμας εμφανίζονται όλα τα προϊόντα που αυτή τη στιγμή υπάρχουν στο κατάστημα. Τα προϊόντα αυτά τα βρίσκουμε

εκτελώντας ένα ερώτημα στους πίνακες shopstocks και product (για να βρούμε το όνομα του προϊόντος).

Όταν πατήσουμε το submit της φόρμας καλείται το ίδιο servlet για να προσθέσει το προϊόν που έχουμε επιλέξει στο κατάστημα. Για να κάνουμε το διαχωρισμό (πότε καλείται για πρώτη φορά και πότε για να προστεθεί προϊόν) χρησιμοποιούμε τη μεταβλητή session command. Αρχικά η μεταβλητή δεν περιέχει τίποτα και τότε το servlet θα μας εμφανίσει τη φόρμα και τα προϊόντα που υπάρχουν στο κατάστημα. Ταυτόχρονα θα θέσει τη μεταβλητή command στην τιμή add. Επομένως όταν θα κληθεί από τον εαυτό του θα αναγνωρίσει την τιμή και θα ξεκινήσει τη διαδικασία προσθήκης του προϊόντος στο κατάστημα. Η διαδικασία αυτή περιλαμβάνει τα ακόλουθα στάδια:

- Αναζήτηση του προϊόντος στον πίνακα shopstocks για να δούμε αν υπάρχει ήδη στο κατάστημα. Αν υπάρχει δεν το προσθέτουμε ξανά.
- Αναζήτηση του προϊόντος στον πίνακα shopstocks αλλά για την κεντρική αποθήκη. Με την αναζήτηση αυτή βρίσκουμε το στοκ αλλά και το ελάχιστο στοκ για το προϊόν στην κεντρική αποθήκη.
- Αν το στοκ της αποθήκης μπορεί να καλύψει το αίτημα για το νέο κατάστημα τότε:
    - Προσθέτουμε το προϊόν στον πίνακα shopstocks
    - Επειδή η κίνηση αυτή είναι εφοδιασμός καταστήματος, εισάγουμε μια νέα εγγραφή στον πίνακα efodiasmos και στον efodiasmosdetails
    - Ελέγχουμε αν το στοκ της κεντρικής αποθήκης έχει πέσει κάτω από το ελάχιστο στοκ. Αν έχει πέσει τότε βρίσκουμε την ελάχιστη τιμή που μας δίνει κάποιος προμηθευτής για το

συγκεκριμένο προϊόν (πίνακας supply) και κάνουμε παραγγελία τόση ποσότητα όση να γίνει το διπλάσιο του ελάχιστου στοκ (εισαγωγή στον πίνακα order). Κατόπιν, θεωρώντας ότι η παραγγελία θα εκτελεστεί άμεσα, ενημερώνουμε το στοκ του προϊόντος της κεντρικής αποθήκης (πίνακας shopstocks).

3.3.5 DeleteShop

Το συγκεκριμένο servlet χρησιμοποιείται για να επιλέξουμε και να διαγράψουμε το κατάστημα που θέλουμε.

Αρχικά εμφανίζει μια φόρμα (της οποίας το action είναι το ίδιο το servlet) η οποία περιέχει όλα τα καταστήματα που έχουν τιμή valid=1 και υπάρχουν στον πίνακα shop. Με την εμφάνιση της φόρμας θέτουμε τη μεταβλητή command του session στην τιμή delete. Με τον τρόπο αυτό διαχωρίζουμε το πότε έχει κληθεί η φόρμα για πρώτη φορά και πότε έχουμε επιλέξει κατάστημα για διαγραφή.

Όταν καλείται το servlet ξανά, τότε εκτελεί ένα update στον πίνακα shop θέτοντας την τιμή valid σε 0 για το κατάστημα που είχε επιλεχθεί.

3.3.6 ShowShop

Το servlet ShowShop χρησιμοποιείται για να εμφανίσουμε τα προϊόντα του καταστήματος που θα επιλέξουμε. Αρχικά εμφανίζουμε μια φόρμα που περιέχει όλα τα καταστήματα που έχουν valid=1. Η φόρμα αυτή έχει δύο submit button.

Το πρώτο έχει κείμενο Εμφάνιση και χρησιμοποιείται για να εμφανίσει τα προϊόντα και τους εφοδιασμούς του καταστήματος. Το action του συγκεκριμένου button είναι το ίδιο το servlet.

Το δεύτερο submit έχει κείμενο Προσθήκη Προϊόντων και όταν πατηθεί θα καλέσει την AddProductsToShop χρησιμοποιώντας javascript με την onclick action.

3.3.7 SelectShop

Το SelectShop χρησιμοποιείται για να επιλέξουμε κατάστημα και με αυτό το κατάστημα να προχωρήσουμε σε εφοδιασμό ή καταχώρηση πωλήσεων.

Εμφανίζει μια φόρμα με ένα select το οποίο περιέχει όλα τα καταστήματα με valid=1. Η φόρμα περιέχει δύο submit. Το πρώτο submit με κείμενο το Εμφάνιση όταν πατηθεί καλεί το servlet UpdateProducts, ενώ το δεύτερο submit όταν πατηθεί καλεί το servlet Efodiasmos.

3.3.8 UpdateProducts

Χρησιμοποιείται για ενημέρωση πωλήσεων σε ένα κατάστημα. Αρχικά εμφανίζει μια λίστα με όλα τα προϊόντα του καταστήματος και δίπλα σε κάθε προϊόν εμφανίζει ένα πεδίο number στο οποίο συμπληρώνω τις πωλήσεις του αντίστοιχου προϊόντος.

Όταν πατηθεί το submit, η φόρμα καλεί το ίδιο servlet (έχοντας θέση στη μεταβλητή command του session τιμή διαφορετική του κενού). Στη συνέχεια το servlet διαβάσει τις τιμές που έχει στείλει η φόρμα και τις αποθηκεύει σε δύο πίνακες. Ένας πίνακας (ο products) με τα productId των προϊόντων και ο άλλος πίνακας (ο values) με τις τιμές που έχουμε ορίσει.

Για κάθε ένα προϊόν εκτελούμε ένα ερώτημα στη βάση δεδομένων (πίνακας shopstocks) για να βρούμε το στοκ και το ελάχιστο στοκ. Στη συνέχεια αφαιρούμε την ποσότητα που πουλήθηκε από το στοκ και ενημερώνουμε τη βάση δεδομένων με το νέο στοκ. Το επόμενο βήμα είναι να ελέγξουμε αν το νέο στοκ είναι μικρότερο από το ελάχιστο στοκ, οπότε θα εμφανίσουμε ένα ενημερωτικό μήνυμα.

3.3.9 Efodiasmos

Χρησιμοποιείται όταν θέλουμε να κάνουμε εφοδιασμό για ένα κατάστημα. Εμφανίζει μια φόρμα με όλα τα προϊόντα του καταστήματος (από τον πίνακα shopstocks) και ένα πεδίο number στο οποίο θα γράφουμε την ποσότητα που θέλουμε για εφοδιασμό.

Όταν ο χρήστης πατήσει το submit καλείται το ίδιο servlet (ο διαχωρισμός γίνεται με τη βοήθεια της μεταβλητής command του session). Η διαδικασία περιλαμβάνει τα ακόλουθα βήματα:

- Εισαγωγή των δεδομένων στον πίνακα efodiasmos (κωδικός καταστήματος και ημερομηνία).

- Για κάθε ένα προϊόν πριν γίνει ο εφοδιασμός αναζητά το απόθεμα της κεντρικής αποθήκης. Αν το απόθεμα είναι μπορεί να καλύψει τον εφοδιασμό:
    - Ενημερώνει το απόθεμα της κεντρικής αποθήκης
    - Ελέγχει αν το απόθεμα είναι μικρότερο από το ελάχιστο στοκ και αν είναι:
        - Βρίσκει τον προμηθευτή με την μικρότερη τιμή για το συγκεκριμένο προϊόν και καταχωρεί τα στοιχεία της παραγγελίας τόσης ποσότητας όσης να είναι το διπλάσιο της ελάχιστης ποσότητας.
    - Ενημερώνει το υπόλοιπο του καταστήματος με τη νέα ποσότητα και
    - Εισάγει στον πίνακα efodiasmosdetails τα δεδομένα (κωδικό προϊόντος και ποσότητα).

3.3.10 AddProduct.jsp

Η λογική είναι παρόμοια με το AddShop.jsp μόνο που χρησιμοποιείται για την προσθήκη ενός νέου προϊόντος. Εμφανίζεται μια φόρμα με τα στοιχεία του προϊόντος που πρέπει να συμπληρωθούν και όταν πατηθεί το κουμπί submit θα κληθεί το servlet AddProduct.

Όπως και στο AddShop.jsp η επικοινωνία μεταξύ του jsp και του servlet (αλλά και αντίστροφα) γίνεται με μεταβλητές session. Στις μεταβλητές αυτές

κρατάμε τις τιμές που έχει συμπληρώσει ο χρήστης (για να τις εμφανίσουμε στο jsp αν έχει κάνει κάποιο λάθος) καθώς και τα μηνύματα λάθους.

3.3.11 AddProduct

Το συγκεκριμένο servlet δέχεται τις τιμές που έχει συμπληρώσει ο χρήστης για το νέο προϊόν που θέλει να εισάγει. Αφού ελέγξει αν έχουν συμπληρωθεί σωστά (π.χ. το μέγεθος του κατασκευαστή να είναι μεγαλύτερο του 1) δημιουργεί μια σύνδεση με τη βάση δεδομένων και εκτελεί μια εντολή insert στον πίνακα product. Κατόπιν για να βρει το productId που αποδόθηκε από τη βάση δεδομένων στη νέα εγγραφή, εκτελεί ένα select με τα στοιχεία του νέου προϊόντος.

Στη συνέχεια εισάγει το προϊόν αυτό, στον πίνακα shopstocks, με τιμές στοκ και ελάχιστου στοκ που έχουν δοθεί από το χρήστη. Η τιμή στο shopId είναι 0, η οποία αντιστοιχεί στην κεντρική αποθήκη.

Αν ολοκληρωθούν όλα τα ερωτήματα σωστά, κλείνει τη σύνδεση με τη βάση δεδομένων και θέτει στην μεταβλητή session okMessage (η οποία περιέχει τα μηνύματα επιτυχία), το αντίστοιχο μήνυμα.

Αν υπάρξει κάποιο πρόβλημα, είτε στις τιμές που έχει δώσει ο χρήστης είτε κατά την εκτέλεση των ερωτημάτων τότε θέτει σε μεταβλητές session τις τιμές που έχει δώσει ο χρήστης και στη μεταβλητή errorMessage (η οποία περιέχει μηνύματα λάθους) το αντίστοιχο μήνυμα και ξανακαλεί το AddProducts.jsp.

3.3.12 DeleteProduct

Το servlet DeleteProduct καλείται όταν θέλουμε να διαγράψουμε ένα προϊόν. Εμφανίζει αρχικά μια φόρμα που περιέχει ένα select με όλα τα προϊόντα που το πεδίο valid έχει τιμή 1, ταξινομημένα με βάση το όνομά τους. Επίσης θέτει τη μεταβλητή command του session στην τιμή delete.

Όταν πατηθεί το πλήκτρο submit της φόρμας τότε καλεί τον εαυτό του και ελέγχει την τιμή της μεταβλητής command του session. Αν έχει κάποια τιμή, σημαίνει ότι πρέπει να εκτελέσει τη διαγραφή του προϊόντος, οπότε διαβάζει το productId και εκτελεί ένα ερώτημα update στον πίνακα products θέτοντας την τιμή valid=0 για το συγκεκριμένο προϊόν.

3.3.13 ShowProduct

Το servlet ShowProduct εμφανίζει πληροφορίες σχετικά με κάποιο προϊόν. Αρχικά εμφανίζει μια φόρμα που περιέχει ένα select με όλα τα προϊόντα από τον πίνακα product που έχουν τιμή valid=1, ταξινομημένα με βάση το πεδίο name.

Όταν πατηθεί το submit, τότε καλεί τον εαυτό της και ελέγχοντας τη μεταβλητή command του session εμφανίζει κάτω από το select των προϊόντων τις πληροφορίες για το προϊόν που επιλέχθηκε και υπάρχουν στη βάση δεδομένων.

3.3.14 ShowMainShop

Το συγκεκριμένο servlet είναι υπεύθυνο για να εμφανίζει όλα τα προϊόντα που υπάρχουν στην αποθήκη καθώς και τα υπόλοιπά τους.

Αρχικά εκτελεί ένα ερώτημα select στον πίνακα shopstocks και στον πίνακα product (για το όνομα των προϊόντων) για να φέρει όλα τα προϊόντα που υπάρχουν στην κεντρική αποθήκη (shopId=0).

Το αποτέλεσμα του select το εμφανίζει σε ένα πίνακα με τρεις στήλες, όνομα προϊόντος, στοκ και ελάχιστο στοκ για κάθε ένα προϊόν.

Στη συνέχεια εκτελεί ένα ερώτημα στους πίνακες

- Order: για να βρει τις παραγγελίες (κωδικό παραγγελίας, ημερομηνία παραγγελίας και ποσότητα παραγγελίας).
- Supply: τιμή προμήθειας προϊόντος
- Product: το όνομα του προϊόντος
- Supplier: όνομα προμηθευτή.

Τα αποτελέσματα αυτά τα εμφανίζει σε μορφή πίνακα.

3.3.15 AddSupplier.jsp

Έχει παρόμοια λογική με τα άλλα jsp και χρησιμοποιείται για να προσθέσουμε ένα νέο προμηθευτή.

Όταν πατηθεί το πλήκτρο submit τότε καλεί το servlet AddSupplier.

3.3.16 AddSupplier

Το servlet AddSupplier διαβάζει τις τιμές που δέχεται από το jsp και τις ελέγχει. Αν υπάρχει κάποιο πρόβλημα στις τιμές που έχει συμπληρώσει ο χρήστης, τότε θέτει αντίστοιχες μεταβλητές session με τις τιμές αυτές, θέτει την αντίστοιχη μεταβλητή session με το μήνυμα λάθους και καλεί την AddSupplier.jsp.

Αν δεν υπάρχει κάποιο πρόβλημα στις τιμές, εκτελεί ένα ερώτημα insert στον πίνακα supplier. Στη συνέχεια, για να βρούμε το supplierId που έχει δοθεί από τη βάση δεδομένων εκτελούμε ένα ερώτημα select, με τα στοιχεία που μόλις εισαγάγαμε. Το supplierId που βρίσκουμε το θέτουμε σε μια μεταβλητή session και καλούμε το servlet AddProductsToSupplier.

3.3.18 AddProductsToSupplier

Αφού έχουμε προσθέσει ένα νέο προμηθευτή πρέπει να του προσθέσουμε και τα προϊόντα που μας προμηθεύει, όπως και τις τιμές τους. Για το σκοπό αυτό είναι υπεύθυνο το servlet AddProductsToSupplier.

Αρχικά διαβάζει από το session το supplierId που του το έχει στείλει το servlet AddSupplier. Στη συνέχεια εμφανίζει μια φόρμα που το action της είναι το ίδιο το servlet. Τα στοιχεία της φόρμας είναι ένα select με όλα τα προϊόντα που υπάρχουν στη βάση δεδομένων και έχουν valid=1,

ταξινομημένα με βάση το πεδίο name. Επίσης υπάρχει ένα πεδίο number στο οποίο ο χρήστης θα συμπληρώσει την τιμή που θα μας προμηθεύει το επιλεγόμενο προϊόν ο προμηθευτής.

Κάτω από τη φόρμα υπάρχει ένας πίνακας που περιέχει όλα τα προϊόντα που έχουμε προσθέσει μέχρι στιγμής στον προμηθευτή με τις αντίστοιχες τιμές τους. Τα στοιχεία αυτά προέρχονται από ένα ερώτημα select στους πίνακες supply και product (για το όνομα του προϊόντος).

Όταν κληθεί το servlet από το submit της φόρμας, διαβάζει τις τιμές που έχουν περαστεί (productId και τιμή) και εκτελεί ένα ερώτημα select για να ελέγξει αν έχουμε ήδη καταχωρίσει το προϊόν αυτό στον προμηθευτή. Αν το έχουμε καταχωρίσει, τότε δεν το εισάγει ξανά. Αν δεν το έχουμε καταχωρίσει, το εισάγει στον πίνακα supply και συνεχίζει την εμφάνιση της υπόλοιπης σελίδας.

3.3.19 DeleteSupplier

Το servlet DeleteSupplier εμφανίζει μια φόρμα με όλους τους προμηθευτές που έχουν το πεδίο valid=1, ταξινομημένους με βάση το πεδίο name. Το action της φόρμας είναι το ίδιο το servlet.

Όταν πατηθεί το submit, καλείται το ίδιο το servlet, το οποίο αναγνωρίζοντας τη μεταβλητή session command, καταλαβαίνει ότι πρέπει να διαγράψει τον προμηθευτή που έχει επιλεγεί. Γι’ αυτό εκτελεί ένα ερώτημα update, στον πίνακα supplier θέτοντας το πεδίο valid=0.

3.3.20 ShowSupplier

Το συγκεκριμένο servlet δείχνει για κάθε ένα προμηθευτή που επιλέγεται, τις παραγγελίες που έχουν γίνει.

Αρχικά εμφανίζει μια φόρμα η οποία περιέχει ένα select με όλους τους προμηθευτές με το πεδίο valid=1. Το action της φόρμας είναι το ίδιο το servlet.

Όταν πατηθεί το submit, αναγνωρίζει ότι κλήθηκε ξανά και διαβάζει το πεδίο supplierId. Με βάση αυτό εκτελεί ένα ερώτημα select στους πίνακες supply και product (για να βρει το όνομα του προϊόντος) και εμφανίζει σε έναν πίνακα με όλα τα προϊόντα που προσφέρει και τις αντίστοιχες τιμές τους.

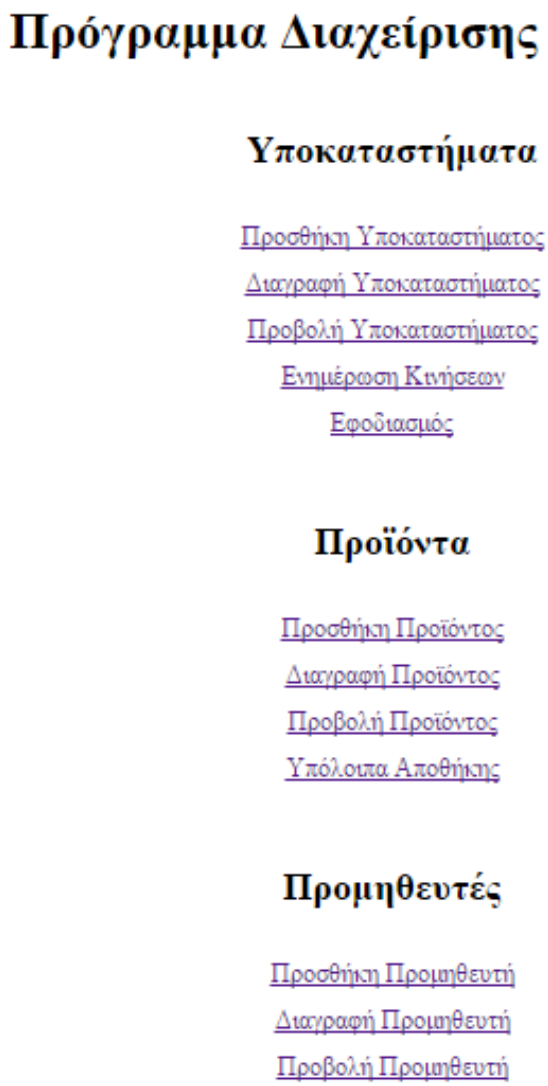
Στη συνέχεια εκτελεί ένα ερώτημα, για να εμφανίσει τις παραγγελίες που του έχουν γίνει, στους πίνακες:

- Order: για κωδικό παραγγελίας, ημερομηνία παραγγελίας και ποσότητα
- Supply: για τιμή χρέωσης
- Product: όνομα προϊόντος
- Supplier: όνομα προμηθευτή

3.4 Τρόπος Λειτουργίας

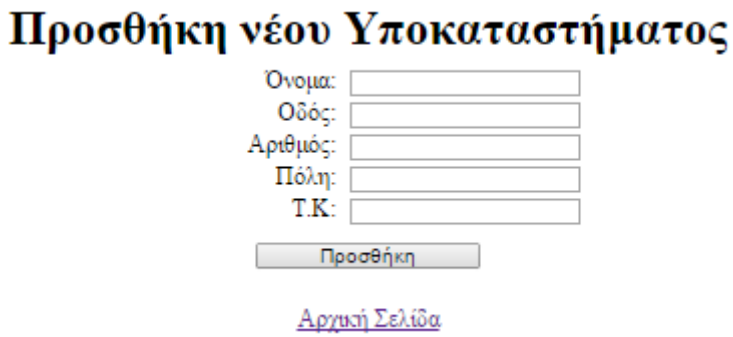
Στην ενότητα αυτή θα παρουσιάσουμε (κυρίως μέσω οθονών) τον τρόπο λειτουργίας της εφαρμογής.

Όπως έχουμε αναφέρει, η πρώτη σελίδα περιέχει σε μορφή υπεσυνδέσμων όλες τις διαθέσιμες δυνατότητες.



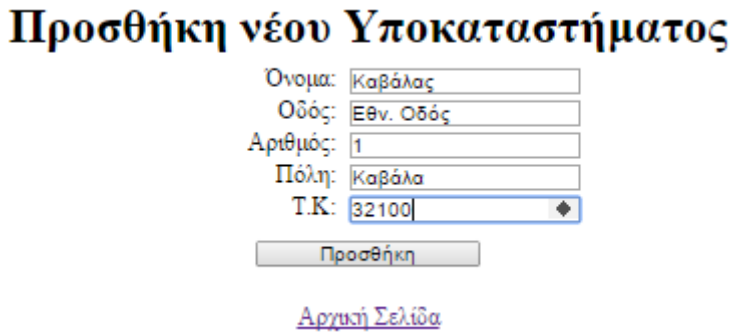
Εικόνα 12: Αρχική Οθόνη Εφαρμογής

Έστω ότι επιλέγουμε αρχικά να εισάγουμε ένα νέο υποκατάστημα. Τότε θα εμφανιστεί η ακόλουθη σελίδα:



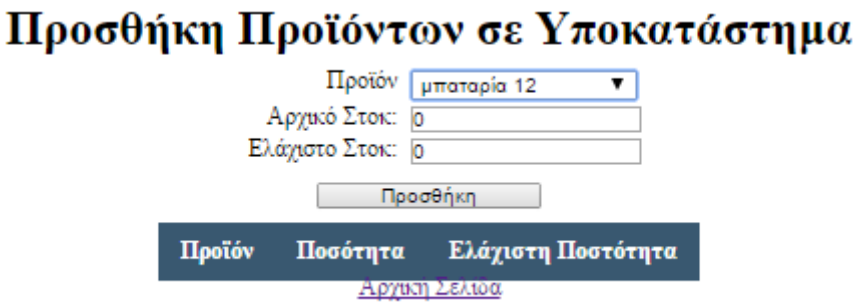
Εικόνα 13: Οθόνη Προσθήκης νέου Υποκαταστήματος

Αφού συμπληρώσουμε τα στοιχεία του υποκαταστήματος που θέλουμε:



Εικόνα 14: Στοιχεία νέου Υποκαταστήματος

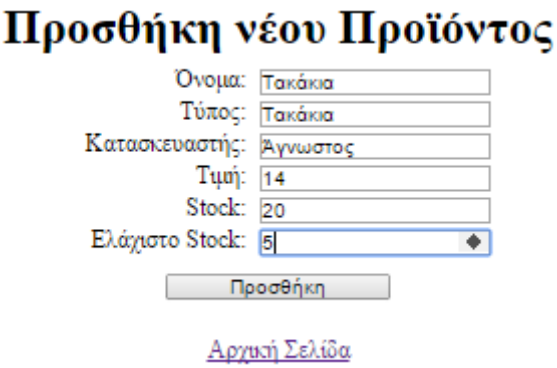
Θα μεταφερθούμε στη σελίδα προσθήκης νέων προϊόντων:



Εικόνα 15: Οθόνη Προσθήκης προϊόντων σε νέο υποκατάστημα

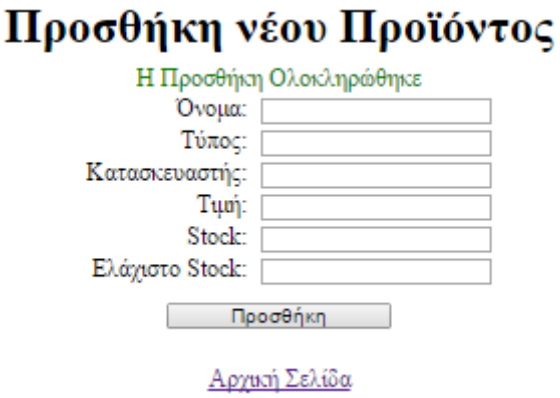
Έστω ότι στο σημείο αυτό δε θέλουμε να προσθέσουμε κάποιο προϊόν στο υποκατάστημα. Οπότε πατώντας το Αρχική Σελίδα θα επιστρέψουμε στην Αρχική Σελίδα της εφαρμογής.

Από εκεί μπορούμε να επιλέξουμε την προσθήκη νέου προϊόντος:



Εικόνα 16: Προσθήκη Προϊόντος

Πατώντας το προσθήκη θα προστεθεί το προϊόν που επιλέξαμε στην κεντρική αποθήκη και θα εμφανιστεί η ακόλουθη οθόνη με το ενημερωτικό μήνυμα:



Εικόνα 17: Ολοκλήρωση Προσθήκης Προϊόντος

Αν πατήσουμε το Υπόλοιπα Αποθήκης θα δούμε μια λίστα με όλα τα προϊόντα που έχει η αποθήκη και τις παραγγελίες που έχουν γίνει. Στη λίστα αυτή θα δούμε και το προϊόν που προσθέσαμε:

Εμφάνιση Κεντρικής Αποθήκης

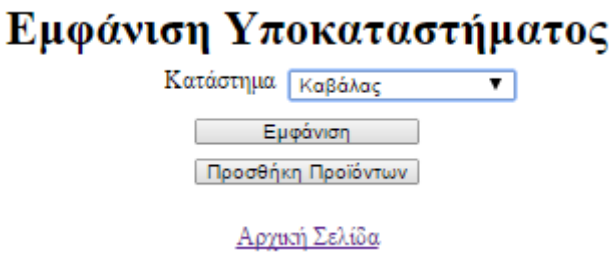
Προϊόν	Ποσότητα	Ελάχιστη Ποσότητα
νέο προϊόν	4	2
ff	13	10
μπαταρία 12	10	5
μπαταρία 16	8	4
λάδι	39	10
Τακάκια	20	5

Κωδικός Παραγγελίας	Ημερομηνία Παραγγελίας	Προϊόν Παραγγελίας	Ποσότητα Παραγγελίας	Προμηθευτής	Τιμή
2	2015-04-08	νέο προϊόν	4	pr1	2.5
3	2015-04-08	νέο προϊόν	4	a	2
4	2015-04-08	νέο προϊόν	3	a	2
5	2015-04-08	νέο προϊόν	3	a	2
6	2015-04-08	νέο προϊόν	4	a	2
7	2015-04-08	νέο προϊόν	3	a	2
8	2015-04-09	μπαταρία 16	6	varta hellas	89
9	2015-04-09	μπαταρία 12	10	varta 2	40

Αρχική Σελίδα

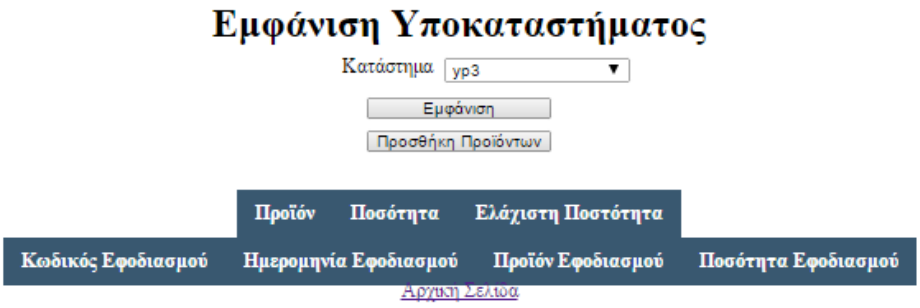
Εικόνα 18: Υπόλοιπα Αποθήκης

Τώρα θέλουμε να προσθέσουμε το προϊόν Τακάκια στο νέο μας υποκατάστημα. Επιλέγουμε από την Αρχική Σελίδα Προβολή Υποκαταστήματος.

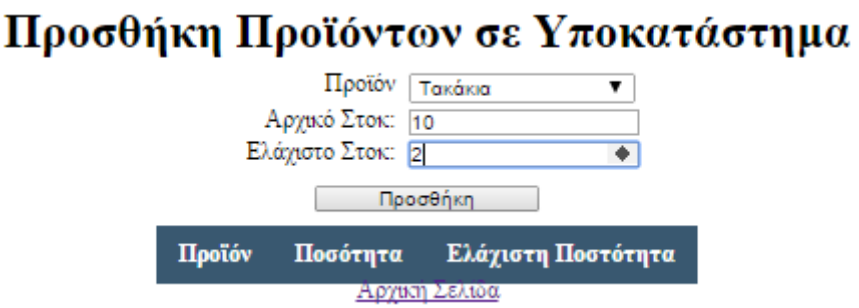


Εικόνα 19: Εμφάνιση Υποκαταστήματος

Και μπορούμε να επιλέξουμε είτε εμφάνιση για να εμφανιστούν τα προϊόντα που έχει είτε προσθήκη για να προσθέσουμε νέα. Αν επιλέξουμε το εμφάνιση:

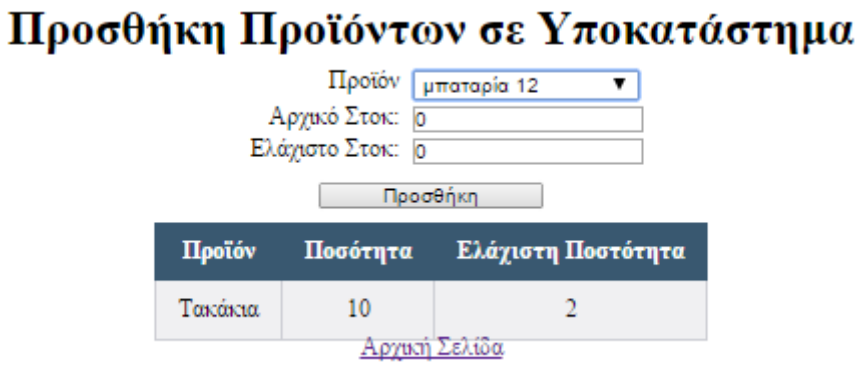


Θα δούμε ότι δεν υπάρχει κανένα προϊόν και δεν έχει γίνει κανένας εφοδιασμός. Ας επιλέξουμε το υποκατάστημά μας και να πατήσουμε το Προσθήκη Προϊόντων.



Εικόνα 20: Προσθήκη προϊόντων σε υποκατάστημα

Επιλέγουμε το προϊόν Τακάκια και το αρχικό και ελάχιστο στοκ. Αν πατήσουμε το προσθήκη, θα δούμε την ακόλουθη εικόνα



Εικόνα 21: Το προϊόν προστέθηκε

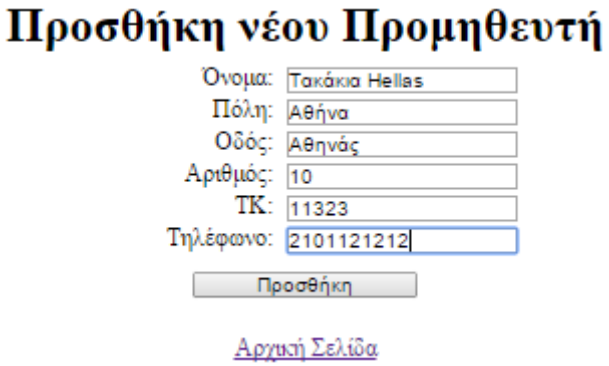
Αν επιλέξουμε Αρχική Σελίδα και επιλέξουμε ξανά το Προβολή Υποκαταστήματος, επιλέξουμε το κατάστημά μας και πατήσουμε το Εμφάνιση θα δούμε το προϊόν και τον Εφοδιασμό που έχει γίνει:

Εικόνα 22: Εμφάνιση Υποκαταστήματος μετά την προσθήκη προϊόντος

Ας ελέγξουμε στο σημείο αυτό τα Υπόλοιπα Αποθήκης, τα οποία πρέπει να έχουν μειωθεί για το συγκεκριμένο προϊόν από 20 σε 10.

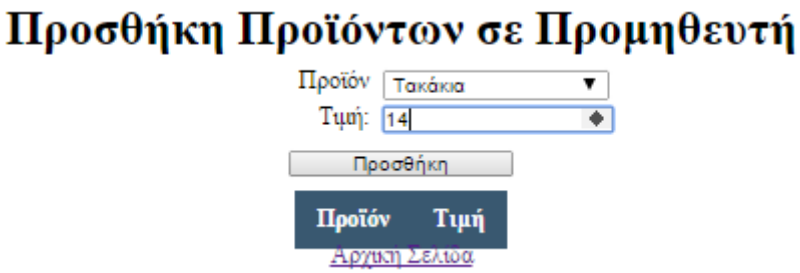
Εικόνα 23: Υπόλοιπα Αποθήκης μετά από προσθήκη προϊόντος σε υποκατάστημα

Στο σημείο αυτό, ας προσθέσω δύο προμηθευτές που μου προμηθεύουν το προϊόν μου. Επιλέγοντας Προσθήκη Προμηθευτή, εισάγω τα στοιχεία του πρώτου προμηθευτή:

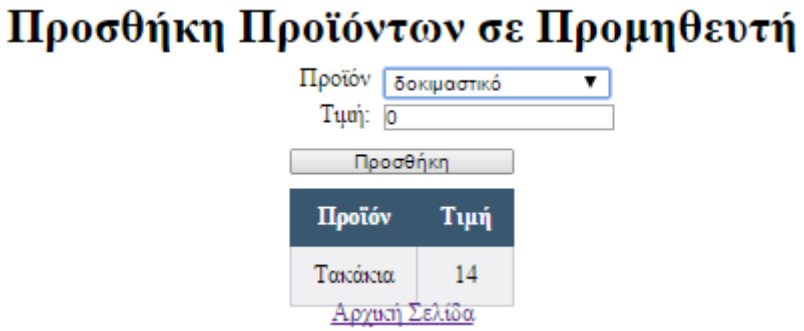


Εικόνα 24: Προσθήκη νέου Προμηθευτή

Η επόμενη οθόνη που θα δω θα είναι για να προσθέσω προϊόντα στον προμηθευτή. Ας προσθέσουμε ένα προϊόν:

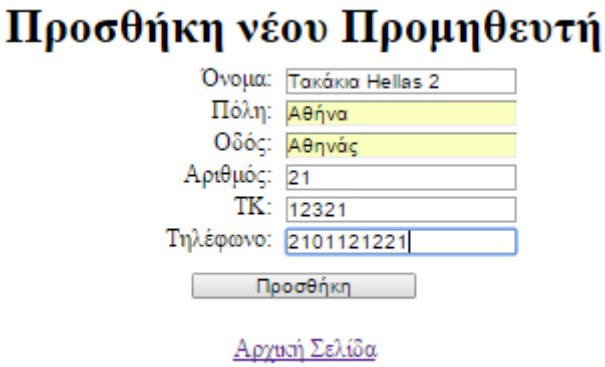


Εικόνα 25: Προσθήκη προϊόντος σε προμηθευτή

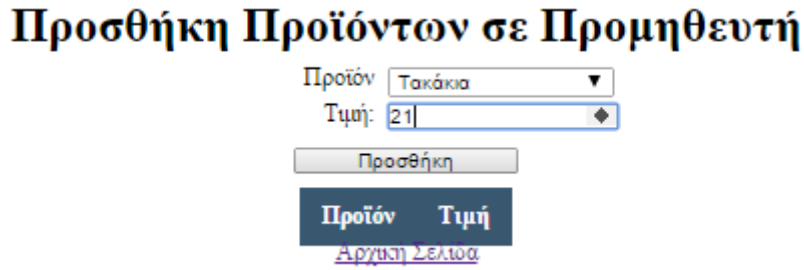


Εικόνα 26: Το προϊόν προστέθηκε στον προμηθευτή

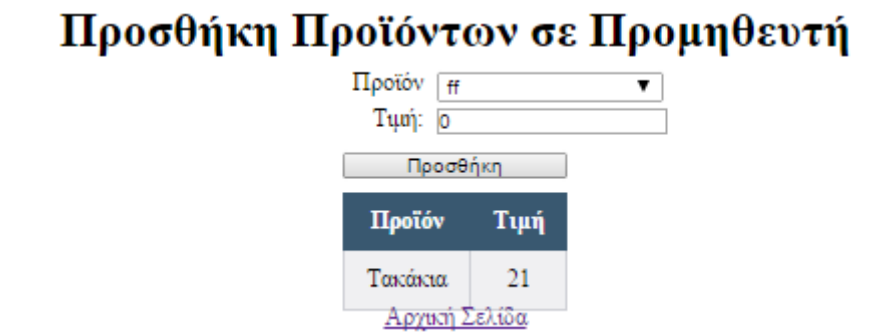
Ας προσθέσουμε ένα προμηθευτή ακόμα, ο οποίος να μας προμηθεύει το ίδιο προϊόν αλλά πιο ακριβά.



Εικόνα 27: Νέος Προμηθευτής

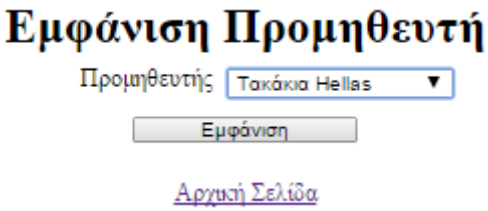


Εικόνα 28: Προσθήκη Προϊόντος σε νέο Προμηθευτή



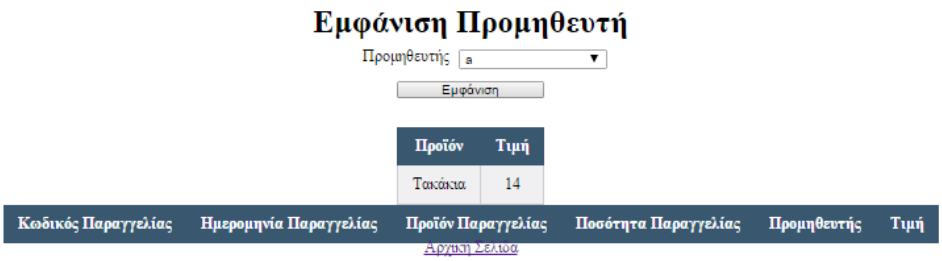
Εικόνα 29: Το προϊόν προστέθηκε στον δεύτερο προμηθευτή

Ας επιλέξουμε τώρα το Προβολή Προμηθευτή. Θα εμφανιστεί η ακόλουθη εικόνα:



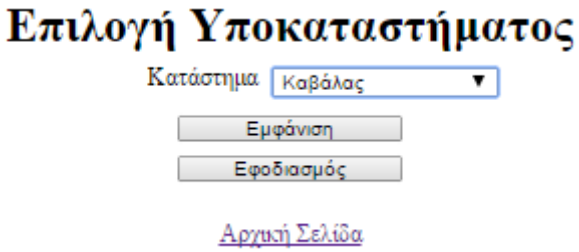
Εικόνα 30: Επιλογή Προμηθευτή για Προβολή

Στην οποία επιλέγουμε τον προμηθευτή που θέλουμε και όταν πατήσουμε εμφάνιση θα δούμε τα προϊόντα που προσφέρει καθώς και τις παραγγελίες που έχουμε κάνει:



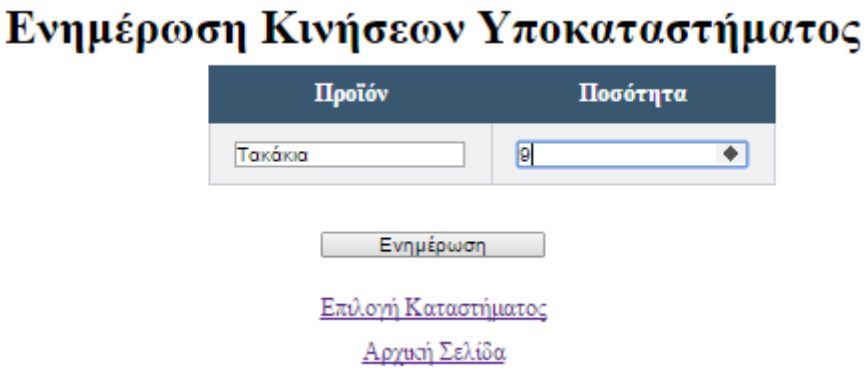
Εικόνα 31: Προβολή Προμηθευτή

Ας υποθέσουμε ότι έγιναν κάποιες πωλήσεις στο υποκατάστημά μας και πρέπει να τις καταχωρίσουμε. Επιλέγουμε Ενημέρωση Κινήσεων:



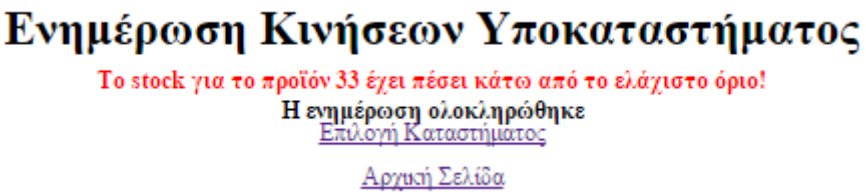
Εικόνα 32: Επιλογή Καταστήματος για καταχώριση κινήσεων

Έστω ότι έγιναν 9 πωλήσεις. Επιλέγουμε το εμφάνιση και καταχωρούμε την ποσότητα και πατάμε το Ενημέρωση.



Εικόνα 33: Προσθήκη Πωλήσεων σε Υποκατάστημα

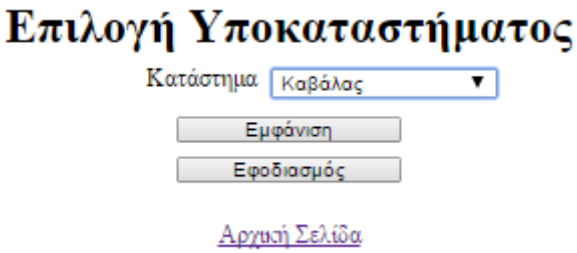
Επειδή η ποσότητα του προϊόντος θα έχει πέσει κάτω από το ελάχιστο στοκ θα εμφανιστεί ένα ενημερωτικό μήνυμα:



Εικόνα 34: Ενημέρωση για ποσότητα λιγότερη από ελάχιστο στοκ

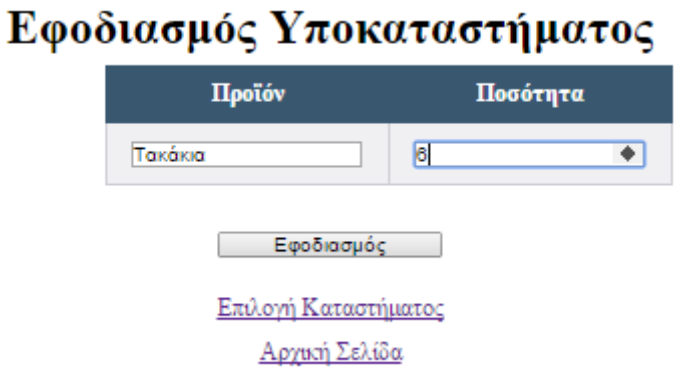
Επομένως πρέπει να προχωρήσουμε στον εφοδιασμό του καταστήματος. Έστω ότι θέλουμε να το εφοδιάσουμε με 6 προϊόντα. Αν γίνει αυτό, επειδή η ποσότητα της κεντρικής αποθήκης θα μειωθεί κάτω από το δικό της ελάχιστο στοκ τότε θα γίνει παραγγελία.

Επιλέγουμε Εφοδιασμός από το αρχικό μενού:



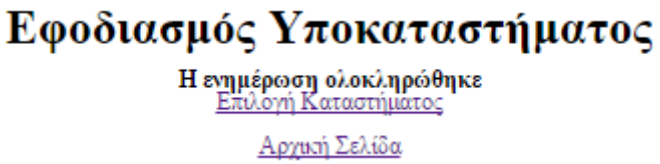
Εικόνα 35: Επιλογή Υποκαταστήματος για Εφοδιασμό

Στη συνέχεια επιλέγουμε το Υποκατάστημα και πατάμε το Εφοδιασμός.



Εικόνα 36: Εφοδιασμός Υποκαταστήματος

Αν πατήσουμε το Εφοδιασμός θα εμφανιστεί ένα ενημερωτικό μήνυμα:



Εικόνα 37: Ενημερωτικό Μήνυμα Εφοδιασμού

Ας δούμε τι έχει γίνει όμως στα υπόλοιπα αποθήκης. Επειδή μειώθηκε η ποσότητα του προϊόντος Τακάκια, πρέπει να έγινε παραγγελία τόσα κομμάτια όσα να γίνουν το διπλάσιο του ελάχιστου στοκ.

Εμφάνιση Κεντρικής Αποθήκης

Προϊόν	Ποσότητα	Ελάχιστη Ποστότητα
νέο προϊόν	4	2
ff	13	10
μπαταρία 12	10	5
μπαταρία 16	8	4
λάδι	39	10
Τακάκια	10	5

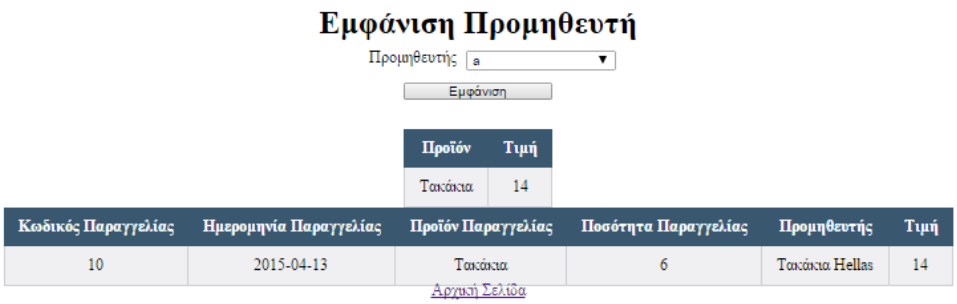
Κωδικός Παραγγελίας	Ημερομηνία Παραγγελίας	Προϊόν Παραγγελίας	Ποσότητα Παραγγελίας	Προμηθευτής	Τιμή
2	2015-04-08	νέο προϊόν	4	pr1	2.5
3	2015-04-08	νέο προϊόν	4	a	2
4	2015-04-08	νέο προϊόν	3	a	2
5	2015-04-08	νέο προϊόν	3	a	2
6	2015-04-08	νέο προϊόν	4	a	2
7	2015-04-08	νέο προϊόν	3	a	2
8	2015-04-09	μπαταρία 16	6	varta hellas	89
9	2015-04-09	μπαταρία 12	10	varta 2	40
10	2015-04-13	Τακάκια	6	Τακάκια Hellas	14

Αρχική Σελίδα

Εικόνα 38: Υπόλοιπα Κεντρικής Αποθήκης

Όπως βλέπουμε τα υπόλοιπα της κεντρικής αποθήκης είναι 10 (διπλάσιο από το υπόλοιπο στοκ). Η παραγγελία έγινε στην Τακάκια Hellas (η μικρότερη τιμή) και το πλήθος ήταν 6. Αρχικά ήταν 10, έγινε εφοδιασμός 6 κομμάτια επομένως έμειναν 4. Για να γίνουν το διπλάσιο του ελάχιστου στοκ θα έπρεπε να παραγγείλουμε 6.

Ας δούμε όμως και την καρτέλα του προμηθευτή.



Εικόνα 39: Καρτέλα Προμηθευτή

Όπως παρατηρούμε υπάρχει η παραγγελία για 6 Τακάκια.

4ο Κεφάλαιο Συμπεράσματα και Μελλοντική Εργασία

4.1 Εισαγωγή - Συμπεράσματα

Με την παρούσα πτυχιακή εργασία κάναμε το πρώτο βήμα για τη δημιουργία ενός πληροφοριακού συστήματος για ανταλλακτικά αυτοκινήτων (αποθήκη) με χρήση Java και servlets. Μετά από την υλοποίηση και τον έλεγχο της εφαρμογής διαπιστώσαμε ότι καλύψαμε πλήρως τις απαιτήσεις που είχαν τεθεί. Οι απαιτήσεις αυτές περιλάμβαναν:

Σε μια βάση δεδομένων θέλουμε να αποθηκεύουμε την παρακάτω πληροφορία για μια αποθήκη ανταλλακτικών.

- Τα στοιχεία κάθε προϊόντος. Όνομα, είδος και κατασκευαστής/παραγωγός και τιμή.
- Τα καταστήματα που προμηθεύει η αποθήκη, Όνομα, διεύθυνση
- Τους προμηθευτές, Ονοματεπώνυμο, διεύθυνση τηλέφωνο
- Τις παραγγελίες προς τους προμηθευτές τις αποθήκης: κωδικό και ποσότητα.
- Τους εφοδιασμούς προς τα καταστήματα. Κάθε εφοδιασμός έχει κωδικό και ημερομηνία.

Επίσης θέλουμε να αποθηκεύουμε:

- Κάθε προμηθευτής συνδέεται με πολλά προϊόντα (τουλάχιστον 1) και κάθε προϊόν παρέχεται από πολλούς προμηθευτές (τουλάχιστον 1). Κάθε προμηθευτής έχει μια τιμή πώλησης η οποία αλλάζει συχνά.

- Μια παραγγελία γίνεται προς ένα και μόνο ένα προμηθευτή.

- Ένας εφοδιασμός αναφέρεται προς ένα και μόνο ένα υποκατάστημα.

4.2 Μελλοντική Εργασία

Όπως αναφέραμε, αυτό ήταν το πρώτο βήμα για την υλοποίηση ενός πληροφοριακού συστήματος για ανταλλακτικά αυτοκινήτων (αποθήκη). Το επόμενο βήμα θα μπορούσε να προσφέρει περισσότερες δυνατότητες, οι οποίες να πλησιάζουν τις δυνατότητες των εμπορικών εφαρμογών όπως:

- Καλύτερη διαχείριση υποκαταστημάτων με δυνατότητα και αφαίρεσης προϊόντων (επιστροφή στην κεντρική αποθήκη)

- Δημιουργία Χρηστών και ρόλων. Θα μπορούσαμε να ορίσουμε χρήστες σε κάθε ένα υποκατάστημα όπως και στην αποθήκη. Στους χρήστες μπορούμε να αποδώσουμε ρόλους όπως

    - διευθυντής αποθήκης (υπεύθυνος για προσθήκη υποκαταστημάτων),

    - διευθυντής καταστήματος (υπεύθυνος για προσθήκη χρηστών στο κατάστημα),

    - χρήστης καταστήματος (υπεύθυνος για την καταχώρηση πωλήσεων και την αποστολή αιτημάτων για εφοδιασμό),

- χρήστης αποθήκης (υπεύθυνος για την καταχώριση νέων παραγγελιών, προσθήκη προϊόντων σε υποκαταστήματα και διαχείριση εφοδιασμών).

- Η καταχώριση των πωλήσεων σε υποκαταστήματα θα μπορούσε να γίνεται με τη χρήση αρχείου όπου θα είναι καταχωρημένα οι κωδικοί των προϊόντων και οι ποσότητες που πουλήθηκαν.

- Οι παραγγελίες θα μπορούσαν να μην εξυπηρετούνται αυτόματα, να υπάρχει ένας χρόνος υλοποίησης ο οποίος θα εξαρτάται από το προϊόν.

ΒΙΒΛΙΟΓΡΑΦΙΑ

DataBase Friends (2014) History of MySQL, http://www.databasefriends.co/2014/02/history-of-mysql.html

Dybka P. (2014) MySQL: Why My? http://www.vertabelo.com/blog/notes-from-the-lab/mysql-history

Greanier T. (2005) Java Εισαγωγή στη Σύγχρονη Τεχνολογία, Αθήνα: Μ. Γκιούρδας

Hall M. & Brown L., 2003, Core Servlets and Javaserver Pages 2nd edition, Prentice Hall

Hunter & Crawford, 1998, Java Servlet Programming, O' Reily, διαθέσιμο από http://docstore.mik.ua/orelly/java-ent/servlet/index.htm

JavaTpoint (n.a) History of Java, http://www.javatpoint.com/history-of-java#

Ostema, R.P. (2004) The History Of Databases, people.cis.ksu.edu/~hankley/d560/Topics/Rogers-Ostema_History.ppt

ΠΑΡΑΡΤΗΜΑΤΑ

Π.1 index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Πρόγραμμα Διαχείρισης Αποθήκης</title>

    <link rel="stylesheet" href="mycss.css">

  </head>

  <body>

    <%session.setAttribute("okMessage", "");

    session.setAttribute("errorMessage", "");%>

    <%session.setAttribute("command", "");%>

    <h1>Πρόγραμμα Διαχείρισης Αποθήκης</h1>

    <h2>Υποκαταστήματα</h2>

    <a class="ahref" href="AddShop.jsp">Προσθήκη Υποκαταστήματος</a><br>

    <a class="ahref" href="DeleteShop">Διαγραφή Υποκαταστήματος</a><br>

    <a class="ahref" href="ShowShop">Προβολή Υποκαταστήματος</a><br>
```

```
        <a class="ahref" href="SelectShop">Ενημέρωση Κινήσεων</a><br>
        <a class="ahref" href="SelectShop">Εφοδιασμός</a><br>
        <h2>Προϊόντα</h2>
        <a class="ahref" href="AddProduct.jsp"> Προσθήκη Προϊόντος</a><br>
        <a class="ahref" href="DeleteProduct">Διαγραφή Προϊόντος</a><br>
        <a class="ahref" href="ShowProduct">Προβολή Προϊόντος</a><br>
        <a class="ahref" href="ShowMainShop"> Υπόλοιπα Αποθήκης</a><br>
        <h2>Προμηθευτές</h2>
        <a class="ahref" href="AddSupplier.jsp"> Προσθήκη Προμηθευτή</a><br>
        <a class="ahref" href="DeleteSupplier">Διαγραφή Προμηθευτή</a><br>
        <a class="ahref" href="ShowSupplier">Προβολή Προμηθευτή</a><br>
    </body>
</html>
```

Π.2 AddSupplier.jsp

```
<%@page contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Προσθήκη Προϊόντος</title>
```

```
    <link rel="stylesheet" href="mycss.css">
</head>
<body>
    <h1>Προσθήκη νέου Προμηθευτή</h1>
    <% String errorMessage=(String)session.getAttribute("errorMessage"); %>
    <% String okMessage=(String)session.getAttribute("okMessage"); %>
    <div style="color: #FF0000;">${errorMessage}</div>
    <div style="color: green;">${okMessage}</div>
    <form method="POST" action="AddSupplier">
    <%
        String name=(String)session.getAttribute("name");
        String city=(String)session.getAttribute("city");
        String street=(String)session.getAttribute("street");
        String number1=(String)session.getAttribute("number1");
        String tk=(String)session.getAttribute("tk");
        String phone=(String)session.getAttribute("phone");
        if (name == null)
            name="";
        if (city == null)
            city="";
```

```
if (street==null)
   street="";
if (number1==null)
   number1="";
if (tk==null)
   tk="";
if (phone==null)
   phone="";
   %>
   <label for="name">Όνομα:</label><input type="text" name="name" maxlength="40" value="<%=name %>"><br>
   <label for="city">Πόλη:</label><input type="text" name="city" maxlength="40" value="<%=city %>"><br>
   <label for="street">Οδός:</label><input type="text" name="street" maxlength="40" value="<%=street %>"><br>
   <label for="number1">Αριθμός:</label><input type="number" name="number1" min=0 step="any" value="<%=number1 %>"><br>
   <label for="tk">TK:</label><input type="number" name="tk" min=0 step="any" value="<%=tk %>"><br>
   <label for="phone">Τηλέφωνο:</label><input type="text" name="phone" min=0 step="any" value="<%=phone %>"><br>
   <input type="submit" value="Προσθήκη" name="submit"><br>
   </form>
```

```
            <a  class="ahref" href="index.jsp"> Αρχική Σελίδα</a>


    </body>
</html>
```

ΙΙ.3 AddShop.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Προσθήκη Υποκαταστήματος</title>
        <link rel="stylesheet" href="mycss.css">
    </head>
    <body>
        <h1>Προσθήκη νέου Υποκαταστήματος</h1>
        <% String errorMessage=(String)session.getAttribute("errorMessage"); %>
        <% String okMessage=(String)session.getAttribute("okMessage"); %>
        <div style="color: #FF0000;">${errorMessage}</div>
```

```
<div style="color: green;">${okMessage}</div>
<form method="POST" action="AddShop">
<%
   String name=(String)session.getAttribute("name");
   String street=(String)session.getAttribute("street");
   String number=(String)session.getAttribute("number");
   String city=(String)session.getAttribute("city");
   String tk=(String)session.getAttribute("tk");
   if (name == null)
      name="";
   if (street == null)
      street="";
   if (number==null)
      number="";
   if (city==null)
      city="";
   if (tk==null)
      tk="";
      %>
      <label for="name">Όνομα:</label><input type="text" name="name"
maxlength="40" value="<%=name %>"><br>
```

```
            <label for="street">Οδός:</label><input type="text" name="street" maxlength="40" value="<%=street %>"><br>

            <label for="number">Αριθμός:</label> <input type="number" name="number" min=0 value="<%=number %>"><br>

            <label for="city">Πόλη:</label> <input type="text" name="city" maxlength="40" value="<%=city %>"><br>

            <label for="tk">T.K:</label> <input type="number" name="tk" min=0 value="<%=tk %>"><br>


            <input type="submit" value="Προσθήκη" name="submit"><br>

            <a class="ahref"  href="index.jsp"> Αρχική Σελίδα</a>

        </form>

    </body>

</html>
```

Π.4 AddProduct.jsp

```
<%@page contentType="text/html; charset=UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>Προσθήκη Προϊόντος</title>
```

```
    <link rel="stylesheet" href="mycss.css">
</head>
<body>
    <h1>Προσθήκη νέου Προϊόντος</h1>
    <% String errorMessage=(String)session.getAttribute("errorMessage"); %>
    <% String okMessage=(String)session.getAttribute("okMessage"); %>
    <div style="color: #FF0000;"><center>${errorMessage}</center></div>
    <div style="color: green;"><center>${okMessage}</center></div>
    <form method="POST" action="AddProduct">
    <%
        String name=(String)session.getAttribute("name");
        String type=(String)session.getAttribute("type");
        String manufacturer=(String)session.getAttribute("manufacturer");
        String price=(String)session.getAttribute("price");
        String stock=(String)session.getAttribute("stock");
        String min_stock=(String)session.getAttribute("min_stock");
        if (name == null)
            name="";
        if (type == null)
            type="";
```

```
if (manufacturer==null)
    manufacturer="";
if (price==null)
    price="";
if (stock==null)
    stock="";
if (min_stock==null)
    min_stock="";
    %>
    <label for="name">Όνομα:</label><input type="text" name="name" maxlength="40" value="<%=name %>"><br>
    <label for="type">Τύπος:</label><input type="text" name="type" maxlength="40" value="<%=type %>"><br>
    <label for="manufacturer">Κατασκευαστής:</label><input type="text" name="manufacturer" maxlength="40" value="<%=manufacturer %>"><br>
    <label for="price">Τιμή:</label><input type="number" name="price" min=0 step="any" value="<%=price %>"><br>
    <label for="stock">Stock:</label><input type="number" name="stock" min=0 step="any" value="<%=stock %>"><br>
    <label for="min_stock">Ελάχιστο Stock:</label><input type="number" name="min_stock" min=0 step="any" value="<%=min_stock %>"><br>
    <input type="submit" value="Προσθήκη" name="submit"><br>
    </form>
```

```
            <a  class="ahref" href="index.jsp"> Αρχική Σελίδα</a>


    </body>
</html>
```

II.5 AddItems.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URLDecoder;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
@WebServlet(urlPatterns = {"/AddItems"})
public class AddItems extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
       ResultSet resultSet;
       String query="";
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           /* TODO output your page here. You may use following sample code. */
```

```
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=UTF-8\">");
            out.println("<title>Προσθήκη Ειδών</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Προσθήκη Ειδών</h1>");

            String fullname=request.getParameter("supplier");

            out.println("<form method=\"post\" action=\"AddOrder\">");

            //String splitName[] = fullname.split("\\s+");
            try {
               myConnection c = new myConnection();
               statement = c.con.createStatement();
```

```
            query="select * from product, supplies where supId="+fullname+" and
product.pname=supplies.pname";
            //out.println(query);

            resultSet = statement.executeQuery(query);
            //out.println("bla:"+resultSet.getInt("lala"));
            out.println("<select name=\"supplier\">");
            while (resultSet.next()){

out.println("<option>"+resultSet.getString("product.pname")+"</option>");
            }

        }
        catch (SQLException ex) {
            out.println(ex.getErrorCode());
            Logger.getLogger(AddItems.class.getName()).log(Level.SEVERE, null,
ex);
        }
        out.println("</select>");
        out.println("Ποσότητα<input type=\"text\" value=\"\"
name=\"posotita\"><br>");
        out.println("<input type=\"submit\" name=\"Submit\" value=\"Επιλογή\">");
```

```
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
```

```
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```

II.6 AddProduct.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
public class AddProduct extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
           throws ServletException, IOException {
       request.setCharacterEncoding("UTF-8");
       HttpSession session = request.getSession();
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {

           String name = request.getParameter("name");
```

```
String type = request.getParameter("type");
String manufacturer = request.getParameter("manufacturer");
String price = request.getParameter("price");
String stock = request.getParameter("stock");
String min_stock = request.getParameter("min_stock");

try {
   double ProductPrice = Double.parseDouble(price);
   if (type.length()<1 || manufacturer.length()<1 || name.length()<1)
   {
      session.setAttribute("name", name);
      session.setAttribute("type", type);
      session.setAttribute("manufacturer", manufacturer);
      session.setAttribute("price", price);
      session.setAttribute("stock", stock);
      session.setAttribute("min_stock", min_stock);
      session.setAttribute("errorMessage", "Πρέπει να συμπληρώσετε τα στοιχεία");
      session.setAttribute("okMessage", "");
      response.sendRedirect("AddProduct.jsp");
   }
```

```
else {

    myConnection c = new myConnection();

    statement = c.con.createStatement();

    String q = "insert into product (name, type, manufacturer, price) values ("'+name+'", "'+type+'", "'+manufacturer+'","'+ProductPrice+")";

    statement.executeUpdate(q);

    q = "select * from product where name="'+name+'" and type="'+type+'" and manufacturer="'+manufacturer+'" and price="'+ProductPrice;

    ResultSet resultSet = statement.executeQuery(q);

    if (resultSet.next()){

       q="insert into shopstocks (shopId, productId, stock, min_stock) values (0, "+resultSet.getInt("productId")+","+stock+","+min_stock+")";

       statement.executeUpdate(q);

    }


    c.con.close();

    session.setAttribute("name", "");

    session.setAttribute("type", "");

    session.setAttribute("manufacturer", "");

    session.setAttribute("price", "");
```

```
        session.setAttribute("stock", "");
        session.setAttribute("min_stock", "");
        session.setAttribute("errorMessage", "");
        session.setAttribute("okMessage", "Η Προσθήκη Ολοκληρώθηκε");
        response.sendRedirect("AddProduct.jsp");
    }
}
catch (NumberFormatException e){

    session.setAttribute("name", name);
    session.setAttribute("type", type);
    session.setAttribute("manufacturer", manufacturer);
    session.setAttribute("price", price);
    session.setAttribute("stock", stock);
    session.setAttribute("min_stock" ,min_stock);
    session.setAttribute("errorMessage", "Η τιμή πρέπει να είναι αριθμός");
    session.setAttribute("okMessage", "");
    response.sendRedirect("AddProduct.jsp");

}
```

```
        } catch (SQLException ex) {
            Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,       null,
ex);
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
```

```
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
```

```
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.7 AddProductsToShop.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;


public class AddProductsToShop extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
           throws ServletException, IOException {


       String q="";
       HttpSession session = request.getSession();
       request.setCharacterEncoding("UTF-8");
```

```
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
   String command = (String)session.getAttribute("command");
   String shopId =(String)session.getAttribute("shopId");
   if (command != null && command.length()>1)
   {
      if (!command.equals("delete")){
      session.setAttribute("command", "");
      myConnection c = new myConnection();
      try {
         int productId = Integer.parseInt(request.getParameter("product"));
         int initialStock = Integer.parseInt(request.getParameter("initialStock"));
         int minStock = Integer.parseInt(request.getParameter("minStock"));

         statement = c.con.createStatement();
         //gia na do an iparxei idi
         //out.println("select * from shopstocks where shopId="+shopId +" and productId="+productId);
         ResultSet resultSet = statement.executeQuery("select * from shopstocks where shopId="+shopId +" and productId="+productId);
```

```
if (!resultSet.next()){
    resultSet.close();
    //gia na do to stock stin kentriki apo8iki
    //out.println("select * from shopstocks where shopId=0 and productId="+productId);
    resultSet = statement.executeQuery("select * from shopstocks where shopId=0 and productId="+productId);
    resultSet.next();


    int min_stock=resultSet.getInt("min_stock");
    int shopstocksid=resultSet.getInt("stockId");
    int centerStock=resultSet.getInt("stock")-initialStock;
    if (centerStock>=0){//an to stock tis kentrikis apo8ikis>=0
      //tote pros8ese to proion oti iparxei sto katastima
      q = "insert into shopstocks (shopId, productId, stock, min_stock) values ("+shopId+", "+productId+", "+initialStock+","+minStock+")";
      //out.println(q);
      statement.executeUpdate(q);

      //enimerose kentriki apo8iki me neo stock
```

```
q="update shopstocks set stock="+centerStock+" where stockId="+shopstocksid;

//out.println(q);

statement.executeUpdate(q);

Calendar c1 = Calendar.getInstance();

int mYear = c1.get(Calendar.YEAR);

int mMonth = c1.get(Calendar.MONTH)+1;

int mDay = c1.get(Calendar.DAY_OF_MONTH);

int lastEfodiasmosId=0;

resultSet.close();

q="insert into efodiasmos (shopId,dateOfEfodiasmos) values ("+shopId+","'+mYear+"-"+mMonth+"-"+mDay+"')";

//out.println(q);

statement.executeUpdate(q);//kano insert ston pinaka efodiasmos

resultSet = statement.executeQuery("select max(eId) as myId from efodiasmos where shopId="+shopId+" and dateOfEfodiasmos='"+mYear+"-"+mMonth+"-"+mDay+"'");

if (resultSet.next()){

   //out.println(resultSet.getInt("myId"));

   lastEfodiasmosId=resultSet.getInt("myId");

}

resultSet.close();
```

```
q="insert into efodiasmosdetails (eId, productId, quantity) values ("+lastEfodiasmosId+","+ productId+","+initialStock+")";

//out.println(q);

statement.executeUpdate(q);//kano insert sto efodiasmosdetails



if (centerStock<=min_stock){

    //kano paraggelia



    min_stock=2*min_stock-centerStock;

    //out.println("select * from supply where productId="+products[i] +" order by price asc");

    resultSet = statement.executeQuery("select * from supply where productId="+productId +" order by price asc");

    if (resultSet.next()){

      q="insert into `order` (dateOfOrder, supplyId,quantity) values ("'+mYear+"-"+mMonth+"-"+mDay+"', "+resultSet.getInt("supplyId")+","+min_stock+")";

//out.println(q);

      //out.println(q);

      statement.executeUpdate(q);

      q="update shopstocks set stock="+(centerStock+min_stock) +" where productId = "+productId+ " and shopId=0";

      //out.println(q);
```

```
                    //out.println(q);
                    statement.executeUpdate(q);
                }
                resultSet.close();
            }

        }
    }
    else
        resultSet.close();
}
catch (SQLException ex) {
    //out.println("Error
"+ex.getLocalizedMessage()+"<br>"+ex.getMessage()+"<br>"+ex.getSQLState());
}
catch (NumberFormatException e){
    q="error";
}
}
else
    shopId=request.getParameter("shop");
```

```
        }
        session.setAttribute("command", "add");
        session.setAttribute("shopId", shopId);
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet AddProductsToShop</title>");
        out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Προσθήκη Προϊόντων σε Υποκατάστημα</h1>");
        out.println("<form method=\"post\" action=\"AddProductsToShop\">");
        out.println("<label for=\"product\">Προϊόν</label><select
name=\"product\">");
        myConnection c = new myConnection();
        try {
            statement = c.con.createStatement();
            //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma
            ResultSet resultSet = statement.executeQuery("select * from product where
valid=1 order by name");
            while (resultSet.next()){
```

```
            String name=resultSet.getString("name");

            out.println("<option
value=\""+resultSet.getString("productId")+"\">"+name+"</option>");

          }

        }

        catch (SQLException ex) {


Logger.getLogger(AddProductsToShop.class.getName()).log(Level.SEVERE,  null,
ex);

        }

        out.println("</select><br>");

        out.println("<label  for=\"initialStock\">Αρχικό  Στοκ:</label><input
type=\"number\" min=0 name=\"initialStock\" value=0><br>");

        out.println("<label  for=\"minStock\">Ελάχιστο  Στοκ:</label><input
type=\"number\" min=0 name=\"minStock\" value=0><br>");

        out.println("<input  type=\"submit\"  value=\"Προσθήκη\"
name=\"Submit\">");

        out.println("</form>");


        try {

            statement = c.con.createStatement();
```

```
            q = "select * from shopstocks, product where shopId = "+shopId +" and
shopstocks.productId=product.productId";

            ResultSet resultSet = statement.executeQuery(q);

            out.println("<table
class=\"center\"><tr><th>Προϊόν</th><th>Ποσότητα</th><th>Ελάχιστη
Ποστότητα</th></tr>");

            while (resultSet.next()){

               out.println("<tr><td>" + resultSet.getString("name")+"</td>");

               out.println("<td>" + resultSet.getString("stock")+"</td>");

               out.println("<td>" + resultSet.getString("min_stock")+"</td>");

               out.println("</tr>");

            }

            out.println("</table>");

         }

         catch (SQLException ex) {//problima stin ektelesi tou erotimatos

            out.println("<h1>Πρόβλημα στην Προσθήκη Προϊόντος!!!!!!!</h1>");

            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");

            out.println("</body>");

            out.println("</html>");

            session.invalidate();

            Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,
null, ex);
```

```
            }




        out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
        out.println("</body>");
        out.println("</html>");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
```

```
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.8 AddProductsToSupplier.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class AddProductsToSupplier extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *

    * @param request servlet request

    * @param response servlet response

    * @throws ServletException if a servlet-specific error occurs

    * @throws IOException if an I/O error occurs

    */

   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

         throws ServletException, IOException {

      String q="";
```

```
HttpSession session = request.getSession();
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
   String command = (String)session.getAttribute("command");
   String supplierId =(String)session.getAttribute("supplierId");
   if (command != null && command.length()>1)
   {
      session.setAttribute("command", "");
      myConnection c = new myConnection();
      try {
         int productId = Integer.parseInt(request.getParameter("product"));
         double price = Double.parseDouble(request.getParameter("price"));

         statement = c.con.createStatement();
         ResultSet resultSet = statement.executeQuery("select * from supply where supplierId="+supplierId +" and productId="+productId);
         if (!resultSet.next()){//na min mporo na pros8eso sto supplier to idio proion
            q = "insert into supply (productId, supplierId, price) values ("+productId+", "+supplierId+", "+price+")";
```

```
            statement.executeUpdate(q);
        }

    }
    catch (SQLException ex) {

    }
    catch (NumberFormatException e){
        q="error";
    }
}
session.setAttribute("command", "add");
session.setAttribute("supplierId", supplierId);
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet AddProductsToSupplier</title>");
out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
out.println("</head>");
out.println("<body>");
```

```
        out.println("<h1>Προσθήκη Προϊόντων σε Προμηθευτή</h1>");

        out.println("<form method=\"post\" action=\"AddProductsToSupplier\">");

        out.println("<label for=\"product\">Προϊόν</label><select
name=\"product\">");

        myConnection c = new myConnection();

        try {

           statement = c.con.createStatement();

           //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma

           ResultSet resultSet = statement.executeQuery("select * from product where
valid=1 order by name");

           while (resultSet.next()){

              String name=resultSet.getString("name");

              out.println("<option
value=\""+resultSet.getString("productId")+"\">"+name+"</option>");

           }

        }

        catch (SQLException ex) {


Logger.getLogger(AddProductsToSupplier.class.getName()).log(Level.SEVERE,
null, ex);

        }

        out.println("</select><br>");
```

```
        out.println("<label for=\"price\">Τιμή:</label><input type=\"number\" min=0 name=\"price\" value=0 step=\"any\"><br>");

        out.println("<input type=\"submit\" value=\"Προσθήκη\" name=\"Submit\">");

        out.println("</form>");



        try {

            statement = c.con.createStatement();



            q = "select * from supply, product where supplierId = "+supplierId +" and supply.productId=product.productId ";

            ResultSet resultSet = statement.executeQuery(q);

            out.println("<table class=\"center\"><tr><th>Προϊόν</th><th>Τιμή</th></tr>");

            while (resultSet.next()){

               out.println("<tr><td>" + resultSet.getString("product.name")+"</td>");

               out.println("<td>" + resultSet.getString("price")+"</td>");

               out.println("</tr>");

            }

            out.println("</table>");

        }
```

```
            catch (SQLException ex) {//problima stin ektelesi tou erotimatos
                out.println("<h1>Πρόβλημα στην Προσθήκη Προϊόντος!!!!!!!</h1>");
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                session.invalidate();
                Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,
null, ex);
            }



        out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
        out.println("</body>");
        out.println("</html>");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
/**
```

```
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
```

```
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Π.9 AddShop.java

```
import java.io.IOException;
```

```
import java.io.PrintWriter;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class AddShop extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *

    * @param request servlet request

    * @param response servlet response
```

```
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String q="";
        request.setCharacterEncoding("UTF-8");
        HttpSession session = request.getSession();
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            String name = request.getParameter("name");
            String street = request.getParameter("street");
            String number = request.getParameter("number");
            String city = request.getParameter("city");
            String tk = request.getParameter("tk");

            if (street.length()<1 || number.length()<1 || city.length()<1 || tk.length()<1)
            {
                session.setAttribute("name", name);
```

```
            session.setAttribute("street", street);
            session.setAttribute("number", number);
            session.setAttribute("city", city);
            session.setAttribute("tk", tk);
            session.setAttribute("errorMessage", "Πρέπει να συμπληρώσετε τα στοιχεία");
            session.setAttribute("okMessage", "");
            response.sendRedirect("AddShop.jsp");
        }
        else {

            myConnection c = new myConnection();
            try {
                statement = c.con.createStatement();
                q = "insert into shop (name, street, number, city, tk) values ("'+name+'", "'+street+'", "+number+","'+city+'","+tk+")";

                statement.executeUpdate(q);
                //gia na bro to shopId
                q = "select * from shop where name="'+name+'" and street="'+street+'" and number="+number+" and city="'+city+"'";
```

```
                ResultSet resultSet = statement.executeQuery(q);
                if (resultSet.next()){
                   String shopId=resultSet.getString("shopId");
                   session.setAttribute("shopId", shopId);//to bazo san session
                }
                c.con.close();
                session.setAttribute("name", "");
                session.setAttribute("type", "");
                session.setAttribute("manufacturer", "");
                session.setAttribute("price", "");
                session.setAttribute("errorMessage", "");
                session.setAttribute("okMessage", "Η Προσθήκη Ολοκληρώθηκε");
                session.setAttribute("command", "");
                response.sendRedirect("AddProductsToShop");//kalo to
addproductstoshop gia na pros8eso proionta sto katastima
            } catch (SQLException ex) {
                session.setAttribute("name", name);
                session.setAttribute("street", street);
                session.setAttribute("number", number);
                session.setAttribute("city", city);
                session.setAttribute("tk", tk);
```

```
                session.setAttribute("errorMessage", "Πρόβλημα στην προσθήκη υποκαταστήματος "+q);
                session.setAttribute("okMessage", "");
                response.sendRedirect("AddShop.jsp");
            }

            /*out.println("Η Προσθήκη Ολοκληρώθηκε");
            out.println("</body>");
            out.println("</html>");*/

            }

        }

    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
```

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Π.10 AddSupplier.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class AddSupplier extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
```

```
    */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        HttpSession session = request.getSession();
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            String name = request.getParameter("name");
            String city = request.getParameter("city");
            String street = request.getParameter("street");
            String number1 = request.getParameter("number1");
            String tk = request.getParameter("tk");
            String phone = request.getParameter("phone");

            try {
                //double ProductPrice = Double.parseDouble(price);
                if (city.length()<1 || street.length()<1 || name.length()<1)
                {
                    session.setAttribute("name", name);
```

```
            session.setAttribute("city", city);
            session.setAttribute("street", street);
            session.setAttribute("number1", number1);
            session.setAttribute("tk", tk);
            session.setAttribute("phone", phone);
            session.setAttribute("errorMessage", "Πρέπει να συμπληρώσετε τα στοιχεία");
            session.setAttribute("okMessage", "");
            response.sendRedirect("AddSupplier.jsp");
        }
        else {
            try{
            myConnection c = new myConnection();
            statement = c.con.createStatement();
            String q = "insert into `supplier` (name, city, street, number, tk, phone) values ("'+name+'", "'+city+'", "'+street+'","'+number1+'","'+tk+'","'+phone+'")";

            //out.println("query="+q);
            statement.executeUpdate(q);//pros8eto supplier
```

```
            q = "select * from supplier where name="'+name+"' and city="'+city+"'
and number="+number1+" and street="'+street+"'";

            //out.println(q);

            ResultSet resultSet = statement.executeQuery(q);

            if (resultSet.next()){

               String supplierId=resultSet.getString("supplierId");

               session.setAttribute("supplierId", supplierId);//to bazo san session

            }

            c.con.close();

            }

            catch (SQLException ex) {

        //Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,    null,
ex);



      }



            session.setAttribute("name", "");

            session.setAttribute("city", "");

            session.setAttribute("street", "");

            session.setAttribute("number1", "");

            session.setAttribute("tk", "");
```

```
            session.setAttribute("phone", "");
            session.setAttribute("errorMessage", "");
            session.setAttribute("okMessage", "Η Προσθήκη Ολοκληρώθηκε");
            response.sendRedirect("AddProductsToSupplier");
        }
    }
    catch (NumberFormatException e){

        session.setAttribute("name", name);
        session.setAttribute("city", city);
        session.setAttribute("street", street);
        session.setAttribute("number1", number1);
        session.setAttribute("tk", tk);
        session.setAttribute("phone", phone);
        session.setAttribute("errorMessage", "Η τιμή πρέπει να είναι αριθμός");
        session.setAttribute("okMessage", "");
        response.sendRedirect("AddSupplier.jsp");

    }
```

```
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
```

```
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
```

```
    }// </editor-fold>


}
```

II.11 DeleteProduct.java

```
import java.io.IOException;

import java.io.PrintWriter;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;



public class DeleteProduct extends HttpServlet {
```

```
    private Statement statement;

    /**

     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

     * methods.

     *

     * @param request servlet request

     * @param response servlet response

     * @throws ServletException if a servlet-specific error occurs

     * @throws IOException if an I/O error occurs

     */

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

            throws ServletException, IOException {

        HttpSession session = request.getSession();

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            String command = (String)session.getAttribute("command");

            if (command != null && command.length()>1)

            {

                session.setAttribute("command", "");
```

```
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet DeleteProduct</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
            myConnection c = new myConnection();
            try {
               statement = c.con.createStatement();
               //kano update tin eggrafi me valid=0
               String q = "update product set valid = 0 where productId = "+request.getParameter("product");;
               statement.executeUpdate(q);
               out.println("<h1>Το Προϊόν διαγράφηκε με επιτυχία!</h1>");
               out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
               out.println("</body>");
               out.println("</html>");
               session.invalidate();
            }
```

```
            catch (SQLException ex) {//problima stin ektelesi tou erotimatos
                out.println("<h1>Πρόβλημα στη Διαγραφή Προϊόντος</h1>");
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                session.invalidate();
                Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE, null, ex);
            }

        }
        else {
            session.setAttribute("command", "delete");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet DeleteProduct</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
```

```
            out.println("<h1>Διαγραφή Προϊόντος</h1>");

            out.println("<form method=\"Post\" action=\"DeleteProduct\">");

            out.println("<label for=\"product\">Προϊόν</label><select
name=\"product\">");

            myConnection c = new myConnection();

            try {

               statement = c.con.createStatement();

               //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma

               ResultSet resultSet = statement.executeQuery("select * from product
where valid=1 order by name");

               while (resultSet.next()){

                  String name=resultSet.getString("name");

                  out.println("<option
value=\""+resultSet.getString("productId")+"\">"+name+"</option>");

               }

            }

            catch (SQLException ex) {

               Logger.getLogger(DeleteProduct.class.getName()).log(Level.SEVERE,
null, ex);

            }

            out.println("</select><br>");
```

```
            out.println("<input type=\"submit\" value=\"Διαγραφή\" name=\"submit\">");
            out.println("<a class=\"ahref\"  href=\"index.jsp\"> Αρχική Σελίδα</a>");
            out.println("</form>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
```

```
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.12 DeleteShop.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class DeleteShop extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *

    * @param request servlet request

    * @param response servlet response

    * @throws ServletException if a servlet-specific error occurs

    * @throws IOException if an I/O error occurs

    */

   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

           throws ServletException, IOException {

       HttpSession session = request.getSession();
```

```
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String command = (String)session.getAttribute("command");
            if (command != null && command.length()>1)
            {
                session.setAttribute("command", "");
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>Servlet DeleteShop</title>");
                out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
                out.println("</head>");
                out.println("<body>");
                myConnection c = new myConnection();
                try {
                    statement = c.con.createStatement();
                    //kano update tin eggrafi me valid=0
                    String q = "update shop set valid = 0 where shopId = "+request.getParameter("shop");;
                    statement.executeUpdate(q);
                    out.println("<h1>Διαγραφή Καταστήματος!!!!!!!</h1>");
```

```
                out.println("<a class=\"ahref\" href=\"index.html\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                session.invalidate();
            }
            catch (SQLException ex) {//problima stin ektelesi tou erotimatos
                out.println("<h1>Πρόβλημα στη Διαγραφή Καταστήματος!!!!!!!</h1>");
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                session.invalidate();
                Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE, null, ex);
            }

        }
        else {
            session.setAttribute("command", "delete");
            out.println("<!DOCTYPE html>");
```

```
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet DeleteShop</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Διαγραφή Καταστήματος</h1>");
            out.println("<form method=\"Post\" action=\"DeleteShop\">");
            out.println("<label for=\"shop\">Κατάστημα</label><select name=\"shop\">");
            myConnection c = new myConnection();
            try {
               statement = c.con.createStatement();
               //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma
               ResultSet resultSet = statement.executeQuery("select * from shop where valid=1 order by name");
               while (resultSet.next()){
                  String name=resultSet.getString("name");
                  out.println("<option value=\""+resultSet.getString("shopId")+"\">"+name+"</option>");
               }
```

```
            }
            catch (SQLException ex) {
                Logger.getLogger(DeleteShop.class.getName()).log(Level.SEVERE,
null, ex);
            }
            out.println("</select><br>");
            out.println("<input type=\"submit\" value=\"Διαγραφή\"
name=\"submit\">");
            out.println("</form><br>");
            out.println("<a class=\"ahref\"  href=\"index.jsp\"> Αρχική Σελίδα</a>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
```

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

ΠΙ.13 DeleteSupplier.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class DeleteSupplier extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *

    * @param request servlet request

    * @param response servlet response

    * @throws ServletException if a servlet-specific error occurs
```

```
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String command = (String)session.getAttribute("command");
            if (command != null && command.length()>1)
            {
                session.setAttribute("command", "");
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>Servlet DeleteSupplier</title>");
                out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
                out.println("</head>");
                out.println("<body>");
                myConnection c = new myConnection();
                try {
```

```
            statement = c.con.createStatement();

            //kano update tin eggrafi me valid=0

            String q = "update supplier set valid = 0 where supplierId = "+request.getParameter("supplier");;

            statement.executeUpdate(q);

            out.println("<h1>Ο Προμηθευτήςδιαγράφηκε με επιτυχία!</h1>");

            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");

            out.println("</body>");

            out.println("</html>");

            session.invalidate();

        }

        catch (SQLException ex) {//problima stin ektelesi tou erotimatos

            out.println("<h1>Πρόβλημα στη Διαγραφή Προμηθευτή</h1>");

            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");

            out.println("</body>");

            out.println("</html>");

            session.invalidate();

            Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE, null, ex);

        }
```

```
        }
        else {
            session.setAttribute("command", "delete");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet DeleteSupplier</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Διαγραφή Προμηθευτή</h1>");
            out.println("<form method=\"Post\" action=\"DeleteSupplier\">");
            out.println("<label for=\"supplier\">Προμηθευτής</label><select
name=\"supplier\">");
            myConnection c = new myConnection();
            try {
                statement = c.con.createStatement();
                //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma
                ResultSet resultSet = statement.executeQuery("select * from supplier
where valid=1 order by name");
```

```
                while (resultSet.next()){
                    String name=resultSet.getString("name");
                    out.println("<option
value=\""+resultSet.getString("supplierId")+"\">"+name+"</option>");
                }
            }
            catch (SQLException ex) {
                Logger.getLogger(DeleteSupplier.class.getName()).log(Level.SEVERE,
null, ex);
            }
            out.println("</select><br>");
            out.println("<input type=\"submit\" value=\"Διαγραφή\"
name=\"submit\">");
            out.println("<a class=\"ahref\"  href=\"index.jsp\"> Αρχική Σελίδα</a>");
            out.println("</form>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

```
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
```

```
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

ΠΙ.14 Efodiasmos.java

```
import java.io.IOException;

import java.io.PrintWriter;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Calendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class Efodiasmos extends HttpServlet {

    private Statement statement;

    /**
```

```
    * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
       boolean valid=true;

       HttpSession session = request.getSession();

       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           String command = (String)session.getAttribute("command");

           int shopId = Integer.parseInt(request.getParameter("shop"));
```

```
myConnection c = new myConnection();
if (command != null && command.length()>1)//edo 8a kano update tis posotites
{
   String[] products=request.getParameterValues("productId");
   String[] values=request.getParameterValues("sale");
   out.println("<!DOCTYPE html>");
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Εφοδιασμός Υποκαταστήματος</title>");
   out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
   out.println("</head>");
   out.println("<body>");
   out.println("<h1>Εφοδιασμός Υποκαταστήματος</h1>");

   try {
      statement = c.con.createStatement();
      //brisko trexousa imerominia
      Calendar c1 = Calendar.getInstance();
      int mYear = c1.get(Calendar.YEAR);
      int mMonth = c1.get(Calendar.MONTH)+1;
```

```
int mDay = c1.get(Calendar.DAY_OF_MONTH);

int lastEfodiasmosId=0;

String q="insert into efodiasmos (shopId,dateOfEfodiasmos) values ("+shopId+","'+mYear+"-"+mMonth+"-"+mDay+"')";

statement.executeUpdate(q);//kano insert ston pinaka efodiasmos

ResultSet resultSet = statement.executeQuery("select max(eId) as myId from efodiasmos where shopId="+shopId+" and dateOfEfodiasmos="'+mYear+"-"+mMonth+"-"+mDay+"'");

if (resultSet.next()){

   //out.println(resultSet.getInt("myId"));

   lastEfodiasmosId=resultSet.getInt("myId");

}

resultSet.close();

for (int i=0; i<products.length;i++)

{

   //brisko stock tis apo8ikis

   resultSet = statement.executeQuery("select * from shopstocks where productId= "+products[i] +" and shopId=0");



   if (resultSet.next()){

      int stock=resultSet.getInt("stock");

      int min_stock=resultSet.getInt("min_stock");
```

```
valid=true;
stock=stock-Integer.parseInt(values[i]);//ipologizo neo stock apo8ikis
if (stock>=0){
q = "update shopstocks set stock ="+ stock+" where productId = "+products[i]+ " and shopId=0";
//out.println(q);
statement.executeUpdate(q);//kano update gia kentriki apo8iki
if (stock<=min_stock){
//kano paraggelia

min_stock=2*min_stock-stock;
resultSet.close();
//out.println("select * from supply where productId="+products[i] +" order by price asc");
resultSet = statement.executeQuery("select * from supply where productId="+products[i] +" order by price asc");
if (resultSet.next()){
q="insert into `order` (dateOfOrder, supplyId,quantity) values ("'+mYear+"-"+mMonth+"-"+mDay+"', "+resultSet.getInt("supplyId")+","+min_stock+")";
//out.println(q);
statement.executeUpdate(q);
```

```
                    q="update shopstocks set stock="+ (stock+min_stock) +" where productId = "+products[i]+ " and shopId=0";
                    //out.println(q);
                    statement.executeUpdate(q);
                }
                resultSet.close();
            }
            else
                resultSet.close();
            resultSet = statement.executeQuery("select * from shopstocks where productId="+products[i] +" and shopId="+shopId);
            if (resultSet.next()){
                stock=resultSet.getInt("stock");

                //String productName = resultSet.getString("product.name");
                stock=stock+Integer.parseInt(values[i]);//ipologizo neo stock
                q = "update shopstocks set stock ="+ stock+" where productId = "+products[i] + " and shopId="+shopId;

                statement.executeUpdate(q);//kano update sto ipokatastima
                q="insert into efodiasmosdetails (eId, productId, quantity) values ("+lastEfodiasmosId+","+ products[i]+","+Integer.parseInt(values[i])+")";
```

```
                    statement.executeUpdate(q);//kano insert sto efodiasmosdetails
                }
            }
            else
            {
                out.println("<center><b>Δυστυχώς ο εφοδιασμός δε μπορεί να συνεχιστεί! Το stock στην κεντρική αποθήκη είναι < 0.</b></center>");
                valid=false;
                break;
            }
        }

    }
    if (valid)
        out.println("<center><b>Η ενημέρωση ολοκληρώθηκε</b></center>");
    }
    catch (SQLException ex) {
        Logger.getLogger(Efodiasmos.class.getName()).log(Level.SEVERE, null, ex);
    }
```

```
            out.println("<a class=\"ahref\" href=\"SelectShop\">Επιλογή Καταστήματος</a><br>");

            out.println("<a class=\"ahref\" href=\"index.jsp\">Αρχική Σελίδα</a>");

            out.println("</body>");

            out.println("</html>");

        }

        else {//emfanizo forma me ola ta product tou katastimatos gia na eisago posotites

            session.setAttribute("command", "command");


        out.println("<!DOCTYPE html>");

        out.println("<html>");

        out.println("<head>");

        out.println("<title>Εφοδιασμός Υποκαταστήματος</title>");

        out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");

        out.println("</head>");

        out.println("<body>");

        out.println("<h1>Εφοδιασμός Υποκαταστήματος</h1>");


        out.println("<form method=\"Post\" action=\"Efodiasmos\">");
```

```
try {
    statement = c.con.createStatement();
    //8a fero ola ta priduct tou katastimatos
    out.println("<input type=\"hidden\" name=\"shop\" value="+shopId+">");
    ResultSet resultSet = statement.executeQuery("select * from shopstocks, product where shopId="+shopId+" and shopstocks.productId=product.productId order by name");
    out.println("<table class=\"center\"><tr><th>Προϊόν</th><th>Ποσότητα</th></tr>");
    while (resultSet.next()){
        String name=resultSet.getString("name");
        out.println("<tr><td><input type=\"text\" value=\""+name+"\" name=\"product\" readonly>");
        //hidden gia na kratao to productId
        out.println("<input type=\"hidden\" value=\""+resultSet.getString("product.productId")+"\" name=\"productId\" ></td>");
        out.println("<td><input type=\"number\" name=\"sale\" min=0 value=0></td></tr>");
    }
}
catch (SQLException ex) {
```

```
            Logger.getLogger(Efodiasmos.class.getName()).log(Level.SEVERE,   null,
ex);

        }

        out.println("</table><br>");

        out.println("<input type=\"submit\" value=\"Εφοδιασμός\"
name=\"submit\">");

        out.println("</form><br>");

        out.println("<a class=\"ahref\" href=\"SelectShop\">Επιλογή
Καταστήματος</a><br>");

        out.println("<a class=\"ahref\" href=\"index.jsp\">Αρχική Σελίδα</a>");

        out.println("</body>");

        out.println("</html>");

        }

    }

}


// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

/**

 * Handles the HTTP <code>GET</code> method.

 *

 * @param request servlet request
```

```
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}// </editor-fold>

}
```

II.15 SelectShop.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class SelectShop extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        response.setContentType("text/html;charset=UTF-8");
        int countproducts=0;
        try (PrintWriter out = response.getWriter()) {
            String command = (String)session.getAttribute("command");
            session.setAttribute("command", "");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Επιλογή Υποκαταστήματος</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Επιλογή Υποκαταστήματος</h1>");
            out.println("<form method=\"Post\" action=\"UpdateProducts\">");
            out.println("<label for=\"shop\">Κατάστημα</label><select name=\"shop\">");
            myConnection c = new myConnection();
```

```
try {
    statement = c.con.createStatement();
    //8a fero ola ta shop me valid=1 taxinomimena me basi to onoma
    ResultSet resultSet = statement.executeQuery("select * from shop where valid=1 order by name");
    while (resultSet.next()){
        countproducts++;
        String name=resultSet.getString("name");
        out.println("<option value=\""+resultSet.getString("shopId")+"\">"+name+"</option>");
    }
}
catch (SQLException ex) {
    Logger.getLogger(SelectShop.class.getName()).log(Level.SEVERE, null, ex);
}
out.println("</select><br>");
if (countproducts>0 ){//an to katastima exei proionta tote 8a emfaniso to submit
    out.println("<input type=\"submit\" value=\"Εμφάνιση\" name=\"submit\">");
    out.println("<input type=\"submit\" value=\"Εφοδιασμός\" onclick=\"form.action='Efodiasmos';\">");
```

```
        }
        out.println("</form><br>");
        out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
        out.println("</body>");
        out.println("</html>");
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

```
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
```

```
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```

Π.16 SelectSupplier.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.URLEncoder;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;


public class SelectSupplier extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
           throws ServletException, IOException {
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           /* TODO output your page here. You may use following sample code. */
           out.println("<!DOCTYPE html>");
```

```
        out.println("<html>");
        out.println("<head>");
        out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=UTF-8\">");
        out.println("<title>Προσθήκη Παραγγελίας</title>");
        out.println("</head>");
        out.println("<body>");
        myConnection c = new myConnection();
        out.println("<h1>Επιλογή Προμηθευτή</h1>");
        out.println("<form method=\"post\" action=\"AddItems\">");
        out.println("<select name=\"supplier\">");
        try {
           statement = c.con.createStatement();
           ResultSet resultSet = statement.executeQuery("select * from supplier");
           while (resultSet.next()){
              String name=resultSet.getString("supname")+"
"+resultSet.getString("supsurname");
              out.println("<option
value=\""+resultSet.getString("supId")+"\">"+name+"</option>");
           }
        }
```

```
            catch (SQLException ex) {
                Logger.getLogger(SelectSupplier.class.getName()).log(Level.SEVERE,
null, ex);
            }
            out.println("</select>");
            out.println("<input type=\"submit\" name=\"Submit\" value=\"Επιλογή\">");
            out.println("</form>");
            try {
                c.con.close();
            } catch (SQLException ex) {
                Logger.getLogger(SelectSupplier.class.getName()).log(Level.SEVERE,
null, ex);
            }
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
    /**
```

```
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
```

```
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.17 ShowMainShop.java

```
import java.io.IOException;

import java.io.PrintWriter;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class ShowMainShop extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *
```

```
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse response)
           throws ServletException, IOException {
       HttpSession session = request.getSession();
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           String command = (String)session.getAttribute("command");
           out.println("<!DOCTYPE html>");
           out.println("<html>");
           out.println("<head>");
           out.println("<title>Εμφάνιση Κεντρικής Αποθήκης</title>");
           out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
           out.println("</head>");
           out.println("<body>");
           out.println("<h1>Εμφάνιση Κεντρικής Αποθήκης</h1>");
           myConnection c = new myConnection();
```

```
try {

    statement = c.con.createStatement();



    String q = "select * from shopstocks, product where shopId = 0 and
shopstocks.productId=product.productId";

    ResultSet resultSet = statement.executeQuery(q);

    out.println("<table
class=\"center\"><tr><th>Προϊόν</th><th>Ποσότητα</th><th>Ελάχιστη
Ποστότητα</th></tr>");

    while (resultSet.next()){

       out.println("<tr><td>" + resultSet.getString("name")+"</td>");

       out.println("<td>" + resultSet.getString("stock")+"</td>");

       out.println("<td>" + resultSet.getString("min_stock")+"</td>");

       out.println("</tr>");

    }

    out.println("</table>");



    resultSet.close();
```

```
q = "select * from `order`, supply, product, supplier where order.supplyId=supply.supplyId and supply.productId=product.productId and supply.supplierId=supplier.supplierId";

resultSet = statement.executeQuery(q);

out.println("<table class=\"center\"><tr><th>Κωδικός Παραγγελίας</th><th>Ημερομηνία Παραγγελίας</th><th>Προϊόν Παραγγελίας</th><th>Ποσότητα Παραγγελίας</th><th>Προμηθευτής</th><th>Τιμή</th></tr>");

while (resultSet.next()){

out.println("<tr><td>" + resultSet.getString("order.orderId")+"</td>");

out.println("<td>" + resultSet.getString("order.dateOfOrder")+"</td>");

out.println("<td>" + resultSet.getString("product.name")+"</td>");

out.println("<td>" + resultSet.getString("order.quantity")+"</td>");

out.println("<td>" + resultSet.getString("supplier.name")+"</td>");

out.println("<td>" + resultSet.getString("supply.price")+"</td>");

out.println("</tr>");

}

out.println("</table>");

}

catch (SQLException ex) {//problima stin ektelesi tou erotimatos
```

```
                out.println("<h1>Πρόβλημα στην Αναζήτηση στη Βάση Δεδομένων!!!!!!!</h1>");
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                session.invalidate();
                Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE, null, ex);
            }

        out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
        out.println("</body>");
        out.println("</html>");
      }
   }

   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
```

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Π.18 ShowProduct.java

```
import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class ShowProduct extends HttpServlet {

   private Statement statement;

   /**

    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>

    * methods.

    *

    * @param request servlet request

    * @param response servlet response

    * @throws ServletException if a servlet-specific error occurs
```

```
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String command = (String)session.getAttribute("command");
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet DeleteProduct</title>");
            out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Προβολή Προϊόντος</h1>");
            out.println("<form method=\"Post\" action=\"ShowProduct\">");
            out.println("<label for=\"product\">Προϊόν</label><select name=\"product\">");
            myConnection c = new myConnection();
```

```
        try {
            statement = c.con.createStatement();
            //8a fero ola ta proionta me valid=1 taxinomimena me basi to onoma
            ResultSet resultSet = statement.executeQuery("select * from product where valid=1 order by name");
            while (resultSet.next()){
                String name=resultSet.getString("name");
                out.println("<option value=\""+resultSet.getString("productId")+"\">"+name+"</option>");
            }
        }
        catch (SQLException ex) {
            Logger.getLogger(ShowProduct.class.getName()).log(Level.SEVERE, null, ex);
        }
        out.println("</select><br>");
        out.println("<input type=\"submit\" value=\"Εμφάνιση\" name=\"submit\">");
        out.println("</form>");

        if (command != null)
        {
```

```
            //session.setAttribute("command", "");


            try {
                statement = c.con.createStatement();
                //kano update tin eggrafi me valid=0
                String q = "select * from product where productId = "+request.getParameter("product");;
                ResultSet resultSet = statement.executeQuery(q);
                if (resultSet.next()){
                    out.println("<table class=\"center\"><tr>");
                    out.println("<th>Όνομα</th>");
                    out.println("<th>Τύπος</th>");
                    out.println("<th>Κατασκευαστής</th>");
                    out.println("<th>Τιμή</th></tr><tr>");
                    out.println("<td>"+resultSet.getString("name")+"</td>");
                    out.println("<td>"+resultSet.getString("type")+"</td>");
                    out.println("<td>"+resultSet.getString("manufacturer")+"</td>");
                    out.println("<td>"+resultSet.getString("price")+"</td>");
                    out.println("</tr></table>");
                }
```

```
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                //session.invalidate();
            }
            catch (SQLException ex) {//problima stin ektelesi tou erotimatos
                out.println("<h1>Πρόβλημα στη Αναζήτηση Προϊόντος!!!!!!!</h1>");
                out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
                out.println("</body>");
                out.println("</html>");
                //session.invalidate();
                Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE, null, ex);
            }

        }
        else {
            session.setAttribute("command", "delete");

            out.println("</body>");
```

```
            out.println("</html>");
        }
    }
}

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
```

```
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.19 ShowShop.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
public class ShowShop extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
           throws ServletException, IOException {
       HttpSession session = request.getSession();
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           String command = (String)session.getAttribute("command");
           out.println("<!DOCTYPE html>");
           out.println("<html>");
```

```
        out.println("<head>");
        out.println("<title>Εμφάνιση Υποκαταστήματος</title>");
        out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Εμφάνιση Υποκαταστήματος</h1>");
        out.println("<form method=\"Post\" action=\"ShowShop\">");
        out.println("<label for=\"shop\">Κατάστημα</label><select name=\"shop\">");
        myConnection c = new myConnection();
        try {
           statement = c.con.createStatement();
           //8a fero ola ta shop me valid=1 taxinomimena me basi to onoma
           ResultSet resultSet = statement.executeQuery("select * from shop where valid=1 order by name");
           while (resultSet.next()){
              String name=resultSet.getString("name");
              out.println("<option value=\""+resultSet.getString("shopId")+"\">"+name+"</option>");
           }
        }
```

```
        catch (SQLException ex) {
            Logger.getLogger(ShowShop.class.getName()).log(Level.SEVERE, null, ex);
        }
        out.println("</select><br>");
        out.println("<input type=\"submit\" value=\"Εμφάνιση\" name=\"submit\">");
        out.println("<input type=\"submit\" value=\"Προσθήκη Προϊόντων\" onclick=\"form.action='AddProductsToShop';\">");
        out.println("</form><br>");
        if (!(command != null && command.length()>1))
        {
            session.setAttribute("command", "delete");
        }
        else {
            try {
                statement = c.con.createStatement();
```

```
String q = "select * from shopstocks, product where shopId = "+request.getParameter("shop") +" and shopstocks.productId=product.productId";

ResultSet resultSet = statement.executeQuery(q);

out.println("<table class=\"center\"><tr><th>Προϊόν</th><th>Ποσότητα</th><th>Ελάχιστη Ποστότητα</th></tr>");

while (resultSet.next()){

   out.println("<tr><td>" + resultSet.getString("name")+"</td>");

   out.println("<td>" + resultSet.getString("stock")+"</td>");

   out.println("<td>" + resultSet.getString("min_stock")+"</td>");

   out.println("</tr>");

}

out.println("</table>");



resultSet.close();

q = "select * from efodiasmos, efodiasmosdetails, product where shopId = "+request.getParameter("shop") +" and efodiasmos.eId=efodiasmosdetails.eId and efodiasmosdetails.productId=product.productId order by dateOfEfodiasmos";

resultSet = statement.executeQuery(q);

out.println("<table class=\"center\"><tr><th>Κωδικός Εφοδιασμού</th><th>Ημερομηνία Εφοδιασμού</th><th>Προϊόν Εφοδιασμού</th><th>Ποσότητα Εφοδιασμού</th></tr>");

while (resultSet.next()){
```

```
                out.println("<tr><td>" +
resultSet.getString("efodiasmos.eId")+"</td>");

                out.println("<td>" +
resultSet.getString("efodiasmos.dateOfEfodiasmos")+"</td>");

                out.println("<td>" + resultSet.getString("product.name")+"</td>");

                out.println("<td>" + resultSet.getString("quantity")+"</td>");

                out.println("</tr>");

            }

            out.println("</table>");

        }

        catch (SQLException ex) {//problima stin ektelesi tou erotimatos

            out.println("<h1>Πρόβλημα στην Αναζήτηση στη Βάση
Δεδομένων!!!!!!!</h1>");

            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");

            out.println("</body>");

            out.println("</html>");

            session.invalidate();

            Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,
null, ex);

        }

    }

    out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
```

```
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
```

```
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

II.20 ShowSupplier.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
public class ShowSupplier extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
           throws ServletException, IOException {
       HttpSession session = request.getSession();
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
           String command = (String)session.getAttribute("command");
           out.println("<!DOCTYPE html>");
           out.println("<html>");
```

```
        out.println("<head>");
        out.println("<title>Εμφάνιση Προμηθευτή</title>");
        out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Εμφάνιση Προμηθευτή</h1>");
        out.println("<form method=\"Post\" action=\"ShowSupplier\">");
        out.println("<label for=\"supplier\">Προμηθευτής</label><select name=\"supplier\">");
        myConnection c = new myConnection();
        try {
           statement = c.con.createStatement();
           //8a fero olous tous supplier taxinomimenous me basi to onoma
           ResultSet resultSet = statement.executeQuery("select * from supplier where valid=1 order by name");
           while (resultSet.next()){
              String name=resultSet.getString("name");
              out.println("<option value=\""+resultSet.getString("supplierId")+"\">"+name+"</option>");
           }
        }
```

```
        catch (SQLException ex) {
            Logger.getLogger(ShowSupplier.class.getName()).log(Level.SEVERE,
null, ex);
        }
        out.println("</select><br>");
        out.println("<input type=\"submit\" value=\"Εμφάνιση\" name=\"submit\">");
        out.println("</form><br>");
        if (!(command != null && command.length()>1))
        {
            session.setAttribute("command", "delete");




        }
        else {
            try {
                statement = c.con.createStatement();


                String q = "select * from supply, product where supplierId =
"+request.getParameter("supplier") +" and supply.productId=product.productId";
                ResultSet resultSet = statement.executeQuery(q);
```

```
out.println("<table
class=\"center\"><tr><th>Προϊόν</th><th>Τιμή</th></tr>");

while (resultSet.next()){

   out.println("<tr><td>" +
resultSet.getString("product.name")+"</td>");

   out.println("<td>" + resultSet.getString("price")+"</td>");

    out.println("</tr>");

}

out.println("</table>");



resultSet.close();

q = "select * from `order`, supply, product, supplier where
order.supplyId=supply.supplyId and supply.productId=product.productId and
supply.supplierId=supplier.supplierId and supply.supplierId =
"+request.getParameter("supplier");

//gia emfanisi paraggelion promi8efti

resultSet = statement.executeQuery(q);

out.println("<table class=\"center\"><tr><th>Κωδικός
Παραγγελίας</th><th>Ημερομηνία Παραγγελίας</th><th>Προϊόν
Παραγγελίας</th><th>Ποσότητα
Παραγγελίας</th><th>Προμηθευτής</th><th>Τιμή</th></tr>");

while (resultSet.next()){

   out.println("<tr><td>" +
resultSet.getString("order.orderId")+"</td>");
```

```
                out.println("<td>" +
resultSet.getString("order.dateOfOrder")+"</td>");
                out.println("<td>" + resultSet.getString("product.name")+"</td>");
                out.println("<td>" + resultSet.getString("order.quantity")+"</td>");
                out.println("<td>" + resultSet.getString("supplier.name")+"</td>");
                out.println("<td>" + resultSet.getString("supply.price")+"</td>");
                out.println("</tr>");
            }
            out.println("</table>");
        }
        catch (SQLException ex) {//problima stin ektelesi tou erotimatos
            out.println("<h1>Πρόβλημα στην Αναζήτηση στη Βάση
Δεδομένων!!!!!!!</h1>");
            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
            out.println("</body>");
            out.println("</html>");
            session.invalidate();
            Logger.getLogger(AddProduct.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
```

```
            out.println("<a class=\"ahref\" href=\"index.jsp\"> Αρχική Σελίδα</a>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
```

```
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

ΠΙ.21 UpdateProducts.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
public class UpdateProducts extends HttpServlet {
   private Statement statement;
   /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException {
       HttpSession session = request.getSession();
       response.setContentType("text/html;charset=UTF-8");
       try (PrintWriter out = response.getWriter()) {
          String command = (String)session.getAttribute("command");
```

```
int shopId = Integer.parseInt(request.getParameter("shop"));
myConnection c = new myConnection();
if (command != null && command.length()>1)//edo 8a kano update tis posotites
{
   String[] products=request.getParameterValues("productId");
   String[] values=request.getParameterValues("sale");
   out.println("<!DOCTYPE html>");
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Ενημέρωση Κινήσεων Υποκαταστήματος</title>");
   out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
   out.println("</head>");
   out.println("<body>");
   out.println("<h1>Ενημέρωση Κινήσεων Υποκαταστήματος</h1>");
   //sto simeio afto 8a ginetai update oi posotites
   try {
      statement = c.con.createStatement();
      for (int i=0; i<products.length;i++)
      {
         //out.println("product:"+products[i]+" value="+values[i]+"<br>");
```

```
//brisko stock kai min_stock gia to proion


ResultSet resultSet = statement.executeQuery("select * from shopstocks where productId="+products[i] +" and shopId="+shopId);
if (resultSet.next()){
int stock=resultSet.getInt("stock");
int min_stock=resultSet.getInt("min_stock");
//String productName = resultSet.getString("product.name");
stock=stock-Integer.parseInt(values[i]);//ipologizo neo stock
String q = "update shopstocks set stock ="+ stock+" where productId = "+products[i]  + " and shopId="+shopId;
//out.println(q);
statement.executeUpdate(q);//kano update
if (stock<=min_stock)//an exei pesei kato apo to min_stock emfanizo minima
out.println("<center><font color=\"red\"><b>To stock για το προϊόν "+ products[i] + " έχει πέσει κάτω από το ελάχιστο όριο!</b></font></center>");
}

}
```

```
            out.println("<center><b>H ενημέρωση ολοκληρώθηκε</b></center>");
        }
        catch (SQLException ex) {

Logger.getLogger(UpdateProducts.class.getName()).log(Level.SEVERE, null, ex);
        }
        out.println("<a class=\"ahref\" href=\"SelectShop\">Επιλογή Καταστήματος</a><br>");
    out.println("<a class=\"ahref\" href=\"index.jsp\">Αρχική Σελίδα</a>");
        out.println("</body>");
        out.println("</html>");
    }
    else {//emfanizo forma me ola ta product tou katastimatos gia na eisago posotites
        session.setAttribute("command", "command");

    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Ενημέρωση Κινήσεων Υποκαταστήματος</title>");
    out.println("<link rel=\"stylesheet\" href=\"mycss.css\">");
```

```
out.println("</head>");

out.println("<body>");

out.println("<h1>Ενημέρωση Κινήσεων Υποκαταστήματος</h1>");


out.println("<form method=\"Post\" action=\"UpdateProducts\">");




try {

    statement = c.con.createStatement();

    //8a fero ola ta priduct tou katastimatos

    out.println("<input type=\"hidden\" name=\"shop\" value="+shopId+">");

    ResultSet resultSet = statement.executeQuery("select * from shopstocks, product where shopId="+shopId+" and shopstocks.productId=product.productId order by name");

    out.println("<table class=\"center\"><tr><th>Προϊόν</th><th>Ποσότητα</th></tr>");

    while (resultSet.next()){

        String name=resultSet.getString("name");

        out.println("<tr><td><input type=\"text\" value=\""+name+"\" name=\"product\" readonly>");

        //hidden gia na kratao to productId
```

```
                out.println("<input type=\"hidden\" value=\""+resultSet.getString("product.productId")+"\" name=\"productId\" ></td>");

                out.println("<td><input type=\"number\" name=\"sale\" min=0 value=0></td></tr>");

            }

        }

        catch (SQLException ex) {

            Logger.getLogger(UpdateProducts.class.getName()).log(Level.SEVERE, null, ex);

        }

        out.println("</table><br>");

        out.println("<input type=\"submit\" value=\"Ενημέρωση\" name=\"submit\">");

        out.println("</form><br>");

        out.println("<a class=\"ahref\" href=\"SelectShop\">Επιλογή Καταστήματος</a><br>");

        out.println("<a class=\"ahref\" href=\"index.jsp\">Αρχική Σελίδα</a>");

        out.println("</body>");

        out.println("</html>");

        }

    }

}
```

```
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
```

```
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
```

```
}
```

II.22 myConnection.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
public class myConnection {
    public Connection con;
    public myConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
```

```
        con=                                                    (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse?useUnicode
=true&characterEncoding=utf-8","root","");
      }
      catch (ClassNotFoundException ex) {
        Logger.getLogger(myConnection.class.getName()).log(Level.SEVERE,  null,
ex);
      }
      catch (SQLException ex) {
        Logger.getLogger(myConnection.class.getName()).log(Level.SEVERE,  null,
ex);
      }
   }
}
```

II.23 mycss.css

```
table {
  border-collapse: separate;
  border-spacing: 0;
}
table.center {
   margin-left:auto;
```

```
    margin-right:auto;
  }

td {
  padding: 10px 15px;
  text-align: center;
}
th {
  background: #395870;
  color: #fff;
  padding: 10px 15px;
  text-align: center;
}
tbody tr:nth-child(even) {
  background: #f0f0f2;
}
td {
  border-bottom: 1px solid #cecfd5;
  border-right: 1px solid #cecfd5;
}
```

```
td:first-child {
  border-left: 1px solid #cecfd5;
}

input, textarea, select  {
  width : 150px;
  padding:0;
  margin:2px
}

form input[type='submit']  {
    display:block;
    margin:10px auto;
}
form {
    width:300px;
    margin:auto;
}
label
{
```

```
    width: 8em;
    float: left;
    text-align: right;
    margin-right: 0.5em;
    display: inline-block;
}

h1 {
    left: 0;
    line-height: 50px;
    margin: auto;
    width: 100%;
    text-align:center;
}
h2 {
    left: 0;
    line-height: 100px;
    margin: auto;
    width: 100%;
    text-align:center;
```

```
}
.ahref {
    display:block;
    text-align:center;
    margin:0 auto;
    line-height: 10px;
}
```