

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.Ι  
ΗΡΑΚΛΕΙΟΥ ΚΡΗΤΗΣ

E-SHOP ΓΙΑ ΥΠΟΛΟΓΙΣΤΕΣ & ΗΛΕΚΤΡΟΝΙΚΑ  
ΕΙΔΗ

---

*Σταυρουλάκης Βασίλειος ΑΜ 3428*

*Ηράκλειο, 2018*

## Table of Contents

ΕΥΧΑΡΙΣΤΙΕΣ.....	5
ABSTRACT .....	6
1. ΠΕΡΙΛΗΨΗ.....	7
2. ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ( Ε – SHOP).....	8
2.1 ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ .....	8
2.2 Η ΛΕΙΤΟΥΡΓΙΑ ΕΝΟΣ ESHOP & ΓΝΩΣΤΕΣ ΠΛΑΤΦΟΡΜΕΣ ΠΑΡΟΧΗΣ ΠΡΟΙΟΝΤΩΝ ΚΑΙ ΥΠΗΡΕΣΙΩΝ .....	10
2.2.1 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ .....	10
2.2.2 AMAZOON & EBAY .....	10
3. ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ NETBEANS & ΛΟΙΠΩΝ ΕΡΓΑΛΕΙΩΝ .....	12
3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA RUNTIME .....	12
3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS IDE .....	12
3.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE.....	13
4. ΠΑΡΟΥΣΙΑΣΗ/RUNNING ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	16
4.1 Η ΠΛΕΥΡΑ ΤΟΥ ADMIN ( ADMIN SIDE ) .....	16
4.2.1 ΕΙΣΟΔΟΣ ΩΣ ADMIN ΣΤΟ ΣΥΣΤΗΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ .....	19
4.2.1.1 ADD PRODUCTS.....	22
4.2.1.2 PRINT PRODUCTS.....	28
4.2.1.3 DELETE PRODUCTS .....	30
4.2.1.4 PRINT CUSTOMERS.....	35
4.2.1.5 ADD COMPANY .....	36
4.2.1.6 REMOVE COMPANY & PRINT COMPANIES.....	39
4.2.1.7 FINISH CUSTOMER ORDER .....	40
4.2.1.8 PRINT FINISHED ORDERS.....	41
4.2.1.9 LOGOUT.....	42
4.3 Η ΠΛΕΥΡΑ ΤΟΥ ΠΕΛΑΤΗ ( CLIENT SIDE ).....	42
4.3.1 PRINT PRODUCTS.....	42
4.3.2 SHOPPING BASKET .....	46
4.3.3 LOGOUT.....	47

5. ΜΕΛΛΟΝΤΙΚΟ ΕΡΓΟ( FUTURE WORK).....	48
5.1 Η ΧΡΗΣΗ ΤΗΣ SQL ΔΙΝΕΙ ΕΥΚΟΛΟΤΕΡΗ ΥΛΟΠΟΙΗΣΗ.....	48
5.2 Η ΣΥΝΔΕΣΗ ΤΗΣ SQL ΜΕ ΤΗΝ JAVA .....	49
5.3 SQL – JAVA ΥΠΕΡΩΤΗΜΑΤΑ(QUERIES) .....	50
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	51
ΑΝΑΦΟΡΕΣ.....	52

## Table of Figures

Figure 1: Η πλατφόρμα Amazon .....	10
Figure 2: Η πλατφόρμα ebay.....	11
Figure 3: Η εγκατάσταση της JAVA.....	12
Figure 4: Η εγκατάσταση της πλατφόρμας NETBEANS 8.2 IDE + JDK .....	13
Figure 5: Αρχικό πάνελ του NETBEANS .....	14
Figure 6: Η επιλογή File του NETBEANS .....	14
Figure 7: Η επιλογή Edit του NETBEANS.....	15
Figure 8: JFrame εισόδου και εγγραφής.....	16
Figure 9: JFrame για εισοδο/εξοδο χρήστη απο σύστημα.....	16
Figure 10: ActionListener() για το κουμπί register .....	17
Figure 11: CheckUser (a).....	17
Figure 12: RegisterClass.....	18
Figure 13: CheckUser (b) .....	19
Figure 14: Admin Panel.....	20
Figure 15: admin main panel (a).....	21
Figure 16: admin main panel (b) .....	21
Figure 17: Add Camera .....	22
Figure 18: Add camera (b) .....	23
Figure 19: Add camera (c) .....	23
Figure 20: Add camera.....	24
Figure 21: Αρχείο cameras.txt πριν την εισαγωγή νέων καμερών .....	24
Figure 22: Φορμα add camera .....	25
Figure 23: Αρχείο cameras.txt μετα την εισαγωγή (κομμάτι αυτού).....	25
Figure 24: Φορμα προσθήκης gaming console .....	26
Figure 25: Αρχείο gaming_consoles πριν την εισαγωγή κονσόλας.....	26
Figure 26: Η προσθήκη μιας κονσόλας στο αρχείο.....	27
Figure 27: Το αρχείο gaming_consoles μετα την προσθήκη κονσόλων .....	27
Figure 28: Φόρτωση αρχείων για εκτύπωση(κομμάτι κώδικα) .....	28

Figure 29: Εκτύπωση συνολικών προϊόντων.....	29
Figure 30: Εκτύπωση προϊόντων (κομματι κώδικα).....	30
Figure 31: Κώδικας μετάβασης για διαγραφή κατάλληλου προϊόντος.....	31
Figure 32: Διαγραφή προϊόντος(κεντρικό πανελ).....	31
Figure 33: Διαγραφή ενός υπολογιστή απο το αρχαιο .....	32
Figure 34: Ανακατασκευή του arrayList pcs_list .....	32
Figure 35: Διαγραφή ποσότητας προϊόντων απο arrayList .....	33
Figure 36: Αποθήκευση της νέας arrayList στο αρχαιο .....	34
Figure 37: Αρχαιο personalComputers πριν τη διαγραφή .....	34
Figure 38: Αρχαιο personalComputers μετα την διαγραφή.....	34
Figure 39: Εκτύπωση πελατών(κομμάτι κώδικα).....	35
Figure 40: Εκτύπωση πελατών .....	36
Figure 41: JFrame προσθήκης εταιρείας στο κατάσταση .....	36
Figure 42: Ανακατασκευή της arrayList των εταιρειων.....	37
Figure 43: Έλεγχος για υπαρξη όμοιας εταιρείας στο σύστημα .....	37
Figure 44: Προσθήκη εταιρείας στο σύστημα .....	38
Figure 45: Αρχαιο εταιρειών πριν την προσθήκη νεας εταιρείας.....	38
Figure 46: Προσθήκη εταιρείας στο αρχαιο .....	39
Figure 47: Αρχαιο μετά την προσθήκη εταιρείας.....	39
Figure 48: Εκτύπωση παραγγελιων.....	42
Figure 49: Δυο τρόποι εκτυπώσεων για τον πελάτη.....	43
Figure 50: Κώδικας υλοποίησης JFrame εκτυπώσεων.....	43
Figure 51: Εκτυπωση προϊόντων ανα ειδος .....	44
Figure 52: Κλήση συναρτήσεων για εκτυπωση διαφορετικών προϊόντων .....	44
Figure 53: Εκτύπωση τηλεοράσεων .....	45
Figure 54: Κώδικας εκτύπωσης τηλεοράσεων .....	45
Figure 55: JFrame για το καλάθι αγορών.....	46
Figure 56: Προσθήκη προϊόντος στο καλάθι αγορών .....	46

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω αρχικά τους γονείς που χωρίς την αγάπη αλλά και την στήριξη τους σε όλα τα χρόνια των σπουδών μου, δεν θα είχα καταφέρει τίποτα στη ζωή μου. Επίσης θέλω να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου Παπαδάκη Νικόλαο, που χωρίς τις πολύτιμες συμβουλές και την καλή διάθεση προς επίλυση αποριών δεν θα έβγαινε εις πέρας η πτυχιακή μου εργασία. Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερα τον Όγλου Γεώργιο, χρόνια φίλο και συμφοιτητή μου για τις πολύτιμες συμβουλές του στο θέμα της γλώσσας προγραμματισμού Java.

## **ABSTRACT**

As technology evolves on a daily basis at a rapid pace, today's man has found many ways to make his life easier in every aspect of his life.

Whether **its** smart devices such as smart phones, or this is a new way of buying and selling products and services with a major profit at the time that is so precious in our times. Now a person can buy and sell via the internet without the need to go to the store of his choice.

It can be very easy to have a good internet connection and little computer knowledge to go to a website that supports an online store and to buy or sell it quickly and quickly as he wishes.

The purpose of this thesis is to introduce to us some e-commerce feature. Also, to give us an understanding of what an e-shop is, and for what reason we must visit a website that has an online store or not.

Then we will try to install the basic tools for developing our software while in the last part we will describe in detail the implementation of an e-shop using the Java programming language.

## 1. ΠΕΡΙΛΗΨΗ

Καθώς η τεχνολογία καθημερινά εξελίσσεται με ραγδαίους ρυθμούς, ο σημερινός άνθρωπος έχει βρεί πάρα πολλούς τρόπους για να κάνει τη ζωή του ευκολότερη σε κάθε τομέα της ζωής του.

Είτε αυτό αφορά έξυπνες συσκευές όπως έξυπνα κινητά είτε αυτό αφορά έναν νέο τρόπο αγοραπωλησίας υπηρεσιών και προϊόντων με κύριο κέρδος τον χρόνο που στην σημερινή εποχή μας είναι τόσο πολύτιμος. Πλέον ένας άνθρωπος μπορεί να αγοράσει και να πουλήσει μέσω διαδικτύου χωρίς να υπάρχει η ανάγκη να μεταβεί στο κατάστημα της αρεσκείας του.

Μπορεί πολύ εύκολα διαθέτωντας μια καλή σύνδεση στο διαδίκτυο και ελάχιστες γνώσεις υπολογιστών να μεταβεί σε μια ιστοσελίδα που υποστηρίζει ένα ηλεκτρονικό κατάστημα και να αγοράζει ή να πουλήσει άμεσα και γρήγορα ότι αυτός επιθυμεί.

Ο σκοπός της πτυχιακής εργασίας είναι να μας παρουσιάζει μερικά χαρακτηριστικά του ηλεκτρονικού εμπορίου, να μας δώσει να καταλάβουμε το τί είναι ένα ηλεκτρονικό κατάστημα και για ποίο λόγο και αν, πρέπει να επισκεφτούμε μια ιστοσελίδα που διαθέτει ένα ηλεκτρονικό κατάστημα ή όχι.

Στη συνέχεια θα προσπαθήσουμε να εγκαταστήσουμε τα βασικά εργαλεία για την ανάπτυξη του λογισμικού μας ενώ στο τελευταίο μέρος θα περιγράψουμε αναλυτικά την υλοποίηση ενός ηλεκτρονικού καταστήματος με χρήση της γλώσσας προγραμματισμού Java.

## 2. ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΚΑΤΑΣΤΗΜΑ( E – SHOP)

### 2.1 ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ

Το ηλεκτρονικό εμπόριο είναι μια διαδικασία κατά την οποία πωλούνται και αγοράζονται προϊόντα και υπηρεσίες διαμέσου του διαδικτύου. Συνήθως οι επιχειρήσεις ηλεκτρονικού εμπορίου απευθύνονται απευθείας στους καταναλωτές μέσω ιστοσελίδων όπως και εφαρμογών για κινητά τηλέφωνα (Wills, 2014)

Στην σημερινή εποχή αν όχι εξ ολοκλήρου ένα μεγάλο μέρος των εταιρειών έχουν στραφεί στο να πωλούν τα προϊόντα ή τις υπηρεσίες τους μέσω του διαδικτύου. Για παράδειγμα ένα ηλεκτρονικό κατάστημα μπορεί να παρέχει προϊόντα κινητής τηλεφωνίας ή ακόμη και υπηρεσίες όπως online αποθηκευτικό χώρο ( cloud services). Μερικές από τις πιο γνωστές πλατφόρμες ηλεκτρονικού εμπορίου είναι το Amazon και το e-Bay (Wienclaw, Ruth, 2013)

Το ηλεκτρονικό εμπόριο έχει τα παρακάτω **θετικά** χαρακτηριστικά που το κάνουν να **ξεχωρίζει** από το απλό «τυπικό» εμπόριο που τόσα χρόνια γνωρίζαμε (Khurana, 2017) :

- Πλέον ένας πελάτης δεν χρειάζεται να μεταβεί στο κατάστημα για να αγοράσει κάτι. Ένα μόνο κλικ αρκεί ώστε η παραγγελία του να έρθει στο σπίτι του εντός ολίγων ημερών.
- Ένας πελάτης μπορεί μέσω της επίσκεψης του στον ιστότοπο του καταστήματος να δει αναλυτικά τα χαρακτηριστικά των προϊόντων που τον ενδιαφέρουν καθώς και εικόνες των προϊόντων. Αυτό βοηθά στην ταυτόχρονη σύγκριση τιμών και χαρακτηριστικών από άλλα καταστήματα, του ίδιου προϊόντος. Το μόνο που χρειάζεται κανείς είναι μια απλή σύνδεση στο διαδίκτυο
- Λόγω του ότι ένα ηλεκτρονικό κατάστημα στεγάζεται στο διαδίκτυο, επόμενο είναι να μη απασχολεί πάρα πολλούς εργαζόμενους με αποτέλεσμα να μπορεί να «ρίξει» τις τιμές ενός προϊόντος ή μιας υπηρεσίας
- Λόγω της ραγδαίας ανάπτυξης της τεχνολογίας, υπάρχουν προχωρημένα συστήματα προγραμματισμού που επιτρέπουν στο ηλεκτρονικό κατάστημα να καταγράφει τις συνήθειες των πελατών με αποτέλεσμα να προτείνει παρόμοια προϊόντα για αγορά χωρίς να χρειαστεί να διερευνήσει μόνος του ολόκληρο τον κατάλογο προϊόντων.



- Ένας πελάτης που θα σκεφτεί να αγοράσει ένα προϊόν ή μια υπηρεσία μπορεί να αφήσει μια κριτική κάτω από το προϊόν. Επομένως ένας επόμενος πελάτης που θα σκεφτεί να αγοράσει το ίδιο προϊόν/υπηρεσία θα μπορέσει να καταλάβει εκ των προτέρων εάν το προϊόν είναι λειτουργικό και αξίο αγοράς πρώτου το αγοράσει.
- Ένα ηλεκτρονικό κατάστημα μπορεί να στείλει τα προϊόντα του σε όλο τον κόσμο όπου και αν αυτός βρίσκεται. Ακόμη και σε διαφορετική ήπειρο.
- Ένα ηλεκτρονικό κατάστημα καθώς «τρέχει» στο διαδίκτυο, μένει ανοικτό εικοσιτέσσερις ώρες το εικοσιτετράωρο
- Πολλές φορές κάποιος χρειάζεται να αλλάξουν το είδος του καταστήματος τους ή να προσθέσουν προϊόντα ή υπηρεσίες. Αυτό δε μπορεί να γίνει εύκολα με ένα κατάστημα που διαθέτει φυσική τοποθεσία! Από την άλλη πλευρά, ένα ηλεκτρονικό κατάστημα μπορεί να αλλάξει το στυλ του (niche) πάρα πολύ εύκολα.

Από την άλλη πλευρά το **ηλεκτρονικό εμπόριο** έχει τα παρακάτω αρνητικά (Khurana, 2018):

- Τα προϊόντα που μπορεί να παραγγείλει κάποιος, δεν έρχονται απευθείας και σε υπερβολικά σύντομο χρονικό διάστημα στην τοποθεσία που όρισε για την παραλαβή αυτών. Πολλές φορές η μεταφορά των προϊόντων μπορεί να αργήσει μέχρι και έναν μήνα εάν για παράδειγμα παραγγείλει κάποιος κάτι από το εξωτερικό.
- Πολλά προϊόντα είναι δύσκολο να επιστραφούν λόγω των μεταφορικών που αναλαμβάνει η εταιρεία για να πάρει πίσω. Ένα τέτοιο παράδειγμα είναι η αγορά επίπλων από το διαδίκτυο.
- Παρόλο που τα προϊόντα διαθέτουν κριτικές κάτω από την περιγραφή, ο κάθε άνθρωπος είναι διαφορετικός, οπότε η κριτική ενός προϊόντος είναι υποκειμενική με αποτέλεσμα το προϊόν που είμαστε διαθέσιμοι να αγοράσουμε να μην ταιριάζει απόλυτα σε εμάς (Chisholm, 1911)
- Ένα μεγάλο θέμα, είναι η ασφάλεια των συναλλαγών. Το ίντερνετ, η γρήγορη εξάπλωση της τεχνολογίας αλλά και η χρήση συνεχώς των πιστωτικών μας καρτών στο διαδίκτυο προκαλεί πολλές φορές «πατήματα» για κακόβουλους χρήστες/ανθρώπους να βρουν τρόπο να υποκλέψουν τους τραπεζικούς μας λογαριασμούς. Οπότε ένα ηλεκτρονικό κατάστημα πρέπει να διαθέτει άριστη προστασία των προσωπικών μας δεδομένων.

## **2.2 Η ΛΕΙΤΟΥΡΓΙΑ ΕΝΟΣ ESHOP & ΓΝΩΣΤΕΣ ΠΛΑΤΦΟΡΜΕΣ ΠΑΡΟΧΗΣ ΠΡΟΙΟΝΤΩΝ ΚΑΙ ΥΠΗΡΕΣΙΩΝ**

### **2.2.1 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ**

Το ηλεκτρονικό κατάστημα δεν είναι τίποτε άλλο από μια ιστοσελίδα που περιέχει προϊόντα και υπηρεσίες διαθέσιμα/ες προς τον πελάτη. Ένας πελάτης για να κάνει είσοδο στο ηλεκτρονικό κατάστημα πρέπει αρχικά να έχει δημιουργήσει ένα λογαριασμό με τα βασικά στοιχεία όπως είναι για παράδειγμα ένα username ένα password και το e-mail του.

Στη συνέχεια το κατάστημα στέλνει ένα e-mail επιβεβαίωσης στο e-mail του. Ο χρήστης πλέον έχει γραφεί στο σύστημα. Στη συνέχεια μόλις κάνει είσοδο στο κατάστημα για την πρώτη φορά πρέπει να συμπληρώσει το προφίλ του, δηλαδή τα βασικά στοιχεία που θα ήταν απαραίτητα για να μπορέσει κάποιος να αγοράσει από το συγκεκριμένο κατάστημα.

Για παράδειγμα θα ζητηθεί ένα τηλέφωνο επικοινωνίας, σίγουρα μια διεύθυνση παραλαβής, και ανάλογα το κατάστημα, πιθανόν και τρόπο πληρωμής (αντικαταβολή ή χρεωστική/πιστωτική κάρτα).

Ένας πελάτης κάθε φορά που θελήσει να αγοράσει κάτι από το ηλεκτρονικό κατάστημα, πολύ απλά θα κάνει είσοδο με το username και το password και στη συνέχεια πηγαίνοντας στην κατάλληλη κατηγορία προϊόντων, βρίσκει το προϊόν που τον ενδιαφέρει και το προσθέτει στο καλάθι αγορών του (και την ποσότητα προϊόντων από το συγκεκριμένο είδος που επέλεξε).

Όταν είναι σίγουρος ότι θέλει να αγοράσει, πατάει επικύρωση και η παραγγελία είναι έτοιμη να φύγει από το κατάστημα. Αρκετές φορές μπορεί να χρειαστεί να καλέσουν τον πελάτη τηλεφωνικά για να επιβεβαιώσει την παραγγελία. Από εκεί και πέρα όλα τα υπόλοιπα τα αναλαμβάνει η εταιρεία ταχυμεταφορών (courier) που συνεργάζεται το εκάστοτε κατάστημα.

### **2.2.2 AMAZOON & EBAY**

Μερικές από τις πιο γνωστές πλατφόρμες πώλησης διαδικτυακών προϊόντων είναι η Amazon και το eBay. Παρακάτω βλέπουμε την κεντρική σελίδα του Amazon (**Figure : 1**)

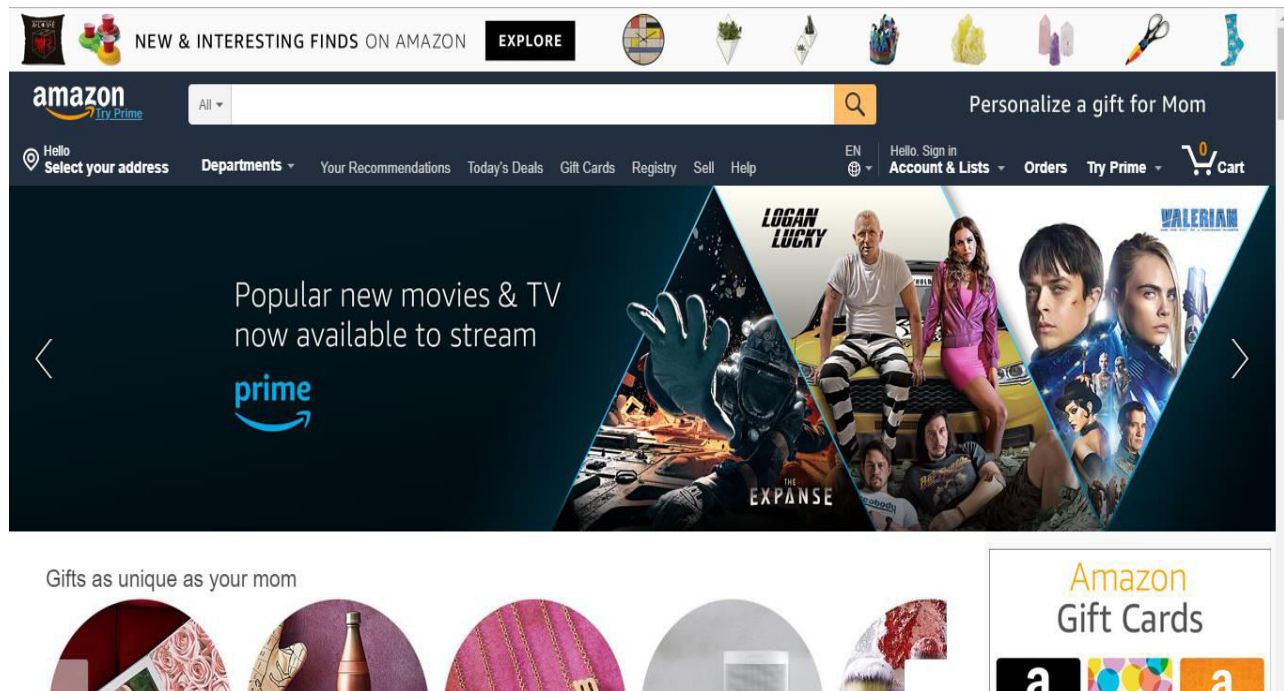


Figure 1: Η πλατφόρμα Amazon

Παρατηρούμε ότι η ιστοσελίδα διαθέτει είσοδο στο σύστημα ώστε ένας πελάτης να εισέλθει μετα προσωπικά του στοιχεία για να μπορέσει να αγοράσει προϊόντα (δεξιά φαίνεται το καλάθι αγορών). Με παρόμοιο τρόπο έχει στηθεί και η πλατφόρμα e-Bay (Figure: 2)

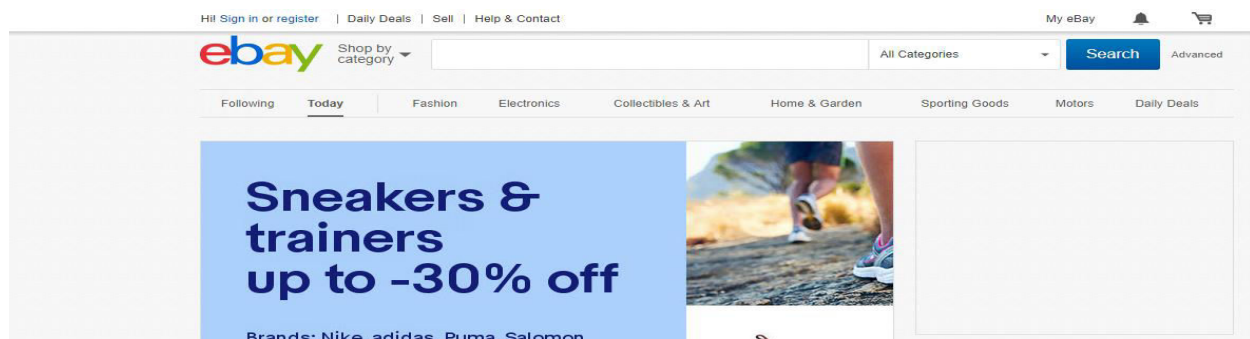


Figure 2: Η πλατφόρμα ebay

### 3. ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ NETBEANS & ΛΟΙΠΩΝ ΕΡΓΑΛΕΙΩΝ

Για να μπορέσουμε να υλοποιήσουμε την εφαρμογή, χρειαζόμαστε το σύστημα μας να τρέχει Java και την πλατφόρμα NETBEANS IDE το οποίο είναι μια πλατφόρμα μέσω της οποίας θα γράψουμε αλλά και θα τρέξουμε την εφαρμογή μας. Η εγκατάσταση των παραπάνω φαίνεται παρακάτω:

#### 3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ JAVA RUNTIME

Αρχικά μεταβαίνουμε στην ιστοσελίδα:

<https://java.com/en/download/>

Και κατεβάζουμε την παρακάτω εφαρμογή(**Figure: 3**)



Figure 3: Η εγκατάσταση της JAVA

#### 3.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS IDE

Στη συνέχεια μεταβαίνουμε στην ιστοσελίδα:

<http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html>

Κατεβάζουμε το NETBEANS 8.2 IDE μαζί με το JDK αφού πρώτα συμφωνήσουμε στους όρους που θέτει η εταιρεία(**Figure: 4**)

**JDK 8u171 with NetBeans 8.2**

This distribution of the JDK includes the Java SE bundle of [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 8u171 and NetBeans 8.2 Cobundle License Agreement](#) to download this software.

Thank you for accepting the [JDK 8u171 and NetBeans 8.2 Cobundle License Agreement](#); you may now download this software.

Java SE and NetBeans Cobundle (JDK 8u171 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	296.49 MB	<a href="#">jdk-8u171-nb-8_2-linux-i586.sh</a>
Linux x64	291.4 MB	<a href="#">jdk-8u171-nb-8_2-linux-x64.sh</a>
Mac OS X x64	363.44 MB	<a href="#">jdk-8u171-nb-8_2-macosx-x64.dmg</a>
Windows x86	327.08 MB	<a href="#">jdk-8u171-nb-8_2-windows-i586.exe</a>
Windows x64	338.65 MB	<a href="#">jdk-8u171-nb-8_2-windows-x64.exe</a>

- [License](#)
- [.Java SF 8 Readme](#)

Figure 4: Η εγκατάσταση της πλατφόρμας NETBEANS 8.2 IDE + JDK

Αφού κατεβάσουμε τα παραπάνω για το δικό μας λειτουργικό σύστημα( εμείς διαθέτουμε Windows και όχι Linux) πατάμε σε όλα ναι και η εγκατάσταση σε 10-12 λεπτά έχει τελειώσει.

Τώρα είμαστε έτοιμοι να ανοίξουμε την εφαρμογή NETBEANS και να γράψουμε κώδικα. Ομως πρώτου προχωρήσουμε στην επεξήγηση του κώδικα που έχουμε γράψει, θα κάνουμε μια μικρή παρουσίαση του τρόπου λειτουργίας της πλατφόρμας NETBEANS.

### 3.3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ NETBEANS 8.2 IDE

Μόλις ανοίξουμε την εφαρμογή βλέπουμε το παρακάτω κεντρικό πανελ( **Figure: 5**)

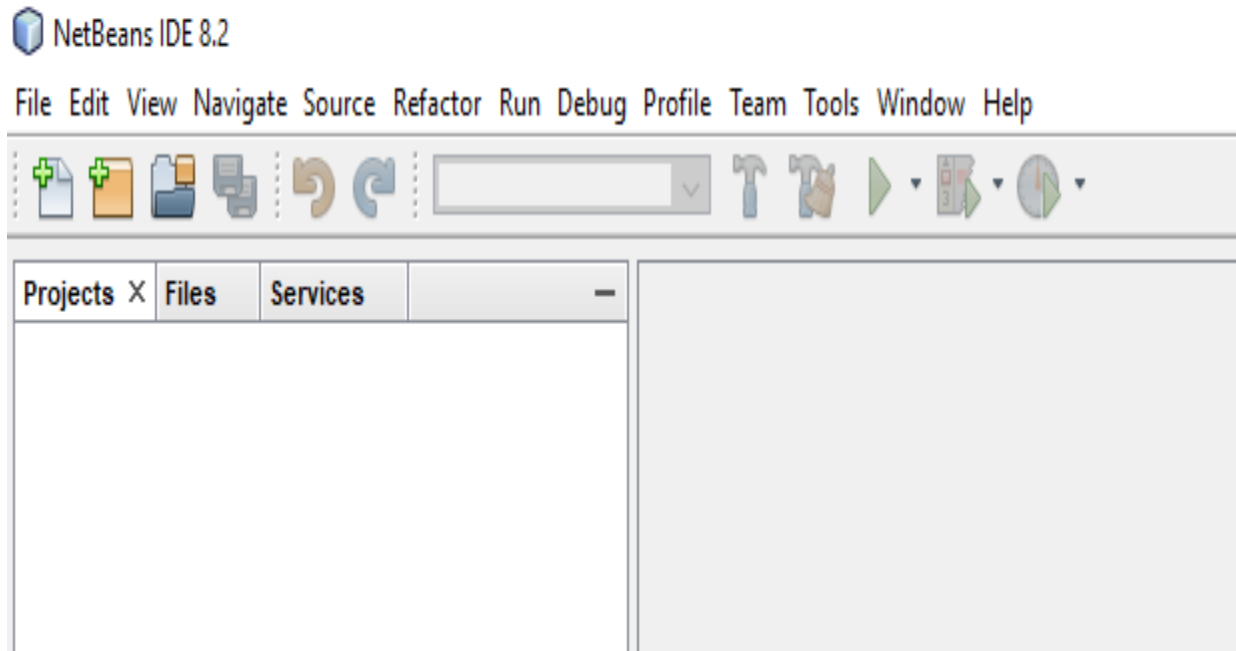


Figure 5: Αρχικό πάνελ του NETBEANS

Στο σημείο projects εμφανίζονται τα project τα οποία έχουμε ανοίξει ενώ στο File μπορούμε να ανοίξουμε ένα ήδη υπάρχον project , να δημιουργήσουμε κάποιο νέο , όπως και το να κλείσουμε κάποιο από τα προτζεκτ μας ( **Figure: 6** )

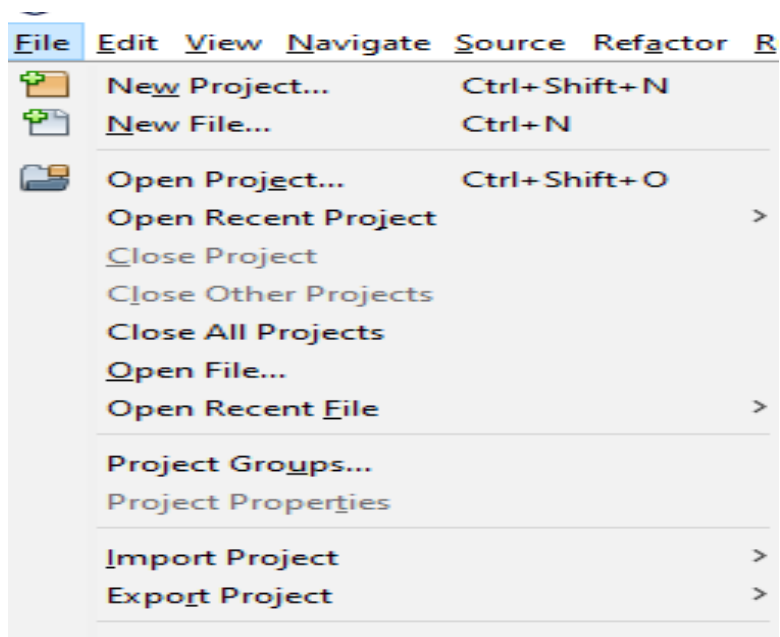


Figure 6: Η επιλογή File του NETBEANS

Με την επιλογή Edit μπορούμε να κανουμε copy/paste/cut τον κώδικα μας οπως και να χρησιμοποιήσουμε την επιλογή Find & Replace για να βρούμε κάτι που χρειαζόμαστε στον κώδικα ή να αντικαταστήσουμε κάποιο μέρος του χωρίς να χρειάζεται να ψάξουμε τον κώδικα γραμμή παρα γραμμή( **Figure: 7** )

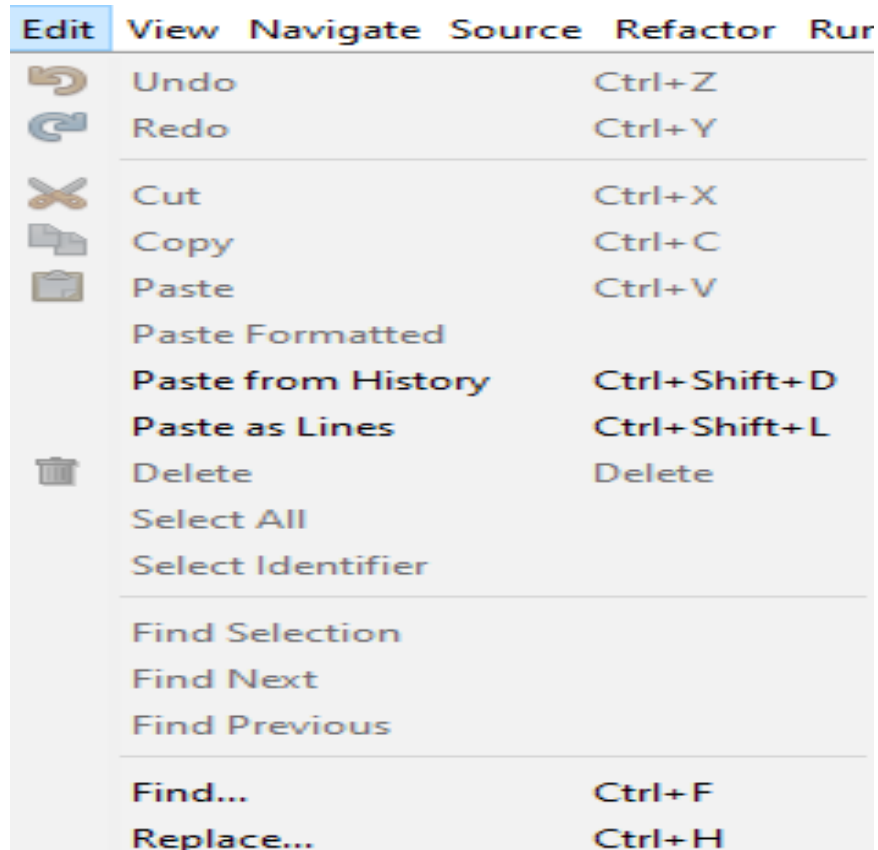


Figure 7: Η επιλογή Edit του NETBEANS

Απο όλες τις υπόλοιπες επιλογές μας αφορά η επιλογή Run μεσω της οποίας τρέχει ο κώδικας. Στο κάτω μέρος του κεντρικού πάνελ του NETBEANS υπάρχει το console στο οποίο εμφανίζονται τα τυχόν λάθη του κώδικα ή η επιτυχής υλοποίηση του.

#### 4. ΠΑΡΟΥΣΙΑΣΗ/RUNNING ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Για να κάνουμε είσοδο στο σύστημα ως admin βάζουμε ως username : vasilis και password : 12345. Αρχικά με το που ανοίξουμε την εφαρμογή μας εμφανίζει ένα JFrame panel που μέσω αυτού μπορούμε είτε να κάνουμε είσοδο είτε εγγραφή ενός νέου χρήστη (**Figure: 8**)

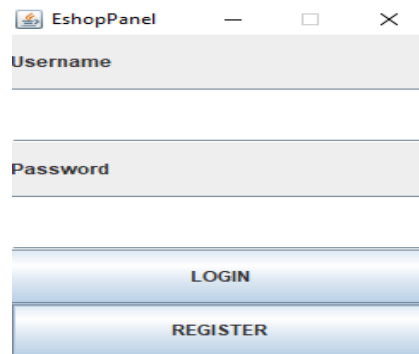


Figure 8: JFrame εισόδου και εγγραφής

Θα δούμε στη συνέχεια τον κώδικα που χρειαστήκαμε για να υλοποιηθεί η είσοδος ενός χρήστη στο σύστημα( **Figure: 9** )

##### 4.1 Η ΠΛΕΥΡΑ ΤΟΥ ADMIN ( ADMIN SIDE )

Μολις κάνουμε είσοδο ως admin εμφανίζεται το παρακάτω JFrame(**Figure: 10**)

```
public class Stavroulakis_Vasilis_Thesis_2018 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("EshopPanel");
        frame.setSize(250, 300);
        frame.setLayout(new BorderLayout());
        JPanel panel = new JPanel();
        frame.add(panel);
        panel.setLayout(new GridLayout(6, 1));
        JLabel label1 = new JLabel("Username");
        JLabel label2 = new JLabel("Password");
        JTextField text1 = new JTextField(20);
        JPasswordField text2 = new JPasswordField(16);
        JButton login = new JButton("LOGIN");
        JButton register = new JButton("REGISTER");
        panel.add(label1);
        panel.add(text1);
        panel.add(label2);
        panel.add(text2);
        panel.add(login);
        panel.add(register);
        register.addActionListener(new RegisterCustomer());
        //login actionListener by getting login values
    }
}
```

Figure 9: JFrame για εισοδο/εξοδο χρήστη απο σύστημα



Βλέπουμε οτι δημιουργήσαμε ένα JFrame, στη συνέχεια ένα JPanel , τα κατάλληλα JTextFields και JLabels και δώσαμε στο κουμπί register έναν ActionListener(). Ο ActionListener του register υλοποιεί αρχικά τα παρακάτω ( **Figure: 10** )

```
login.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        String a1 = text1.getText();
        String a2 = text2.getText();
        try {
            checkUser(a1, a2);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(Stavroulakis_Vasilis_Thesis_2018.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(Stavroulakis_Vasilis_Thesis_2018.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setResizable(false);
```

Figure 10: ActionListener() για το κουμπί register

Βλέπουμε οτι πήραμε τις τιμές των JTextFields που δημιουργήσαμε πιο πάνω δηλαδή τις τιμές username και password και τις περάσαμε σε μια κλάση που την ονομάσαμε checkUser η οποία θα ελέγξει με κατάλληλους τρόπους την είσοδο ενός admin ή customer. Φυσικά τα κατάλληλα exceptions και τις υποχρεωτικές λειτουργίες για να εμφανίζεται το frame στην οθόνη. Ας δούμε το πώς λειτουργεί η κλάση checkUser.

```
public static void checkUser(String a1, String a2) throws FileNotFoundException, IOException {
    //admin's credentials username: vasilis, password: 12345
    String admin_username = "vasilis";
    String admin_password = "12345";
    List<RegisterClass> customers_list = new ArrayList<>();
    Scanner file;
    file = new Scanner(new File("customers.txt"));

    while (file.hasNext()) {
        String line = file.nextLine();
        if (!line.isEmpty()) {
            String[] stringArr = line.split("\t");
            String fname = stringArr[0];
            String surname = stringArr[1];
            String address = stringArr[2];
            String city = stringArr[3];
            String tk = stringArr[4];
            String afm = stringArr[5];
            String username = stringArr[6];
            String password = stringArr[7];
            //toys prosthetoyme sthn lista me oloys tous users toy eshop
            RegisterClass customer = new RegisterClass(fname, surname, address, city, tk, afm, username, password);
            customers_list.add(customer);
        }
    }
}
```

Figure 11: CheckUser (a)

Εδώ αρχικά περνάμε στη κλάση τις τιμές username , password που λάβαμε απο τον χρήστη, στη συνέχεια δημιουργούμε ενα arrayList που είναι σύμφωνα με το RegisterClass και ανοίγουμε το αρχείο customers.txt και μέσω της χρήσης μιας κλάσης while προσπαθούμε να φορτώσουμε στη λίστα customers\_list τα περιεχόμενα του .txt αρχείου. Πρωτού συνεχίσουμε ας δούμε το πώς δημιουργήθηκε η RegisterClass ( **Figure : 12**)

```
public class RegisterClass {

    String fname;
    String surname;
    String address;
    String city;
    String TK;
    String AFM;
    String username;
    String password;

    public RegisterClass(String fname, String surname, String address, String city, String TK, String AFM, String usern
        this.fname = fname;
        this.surname = surname;
        this.address = address;
        this.city = city;
        this.TK = TK;
        this.AFM = AFM;
        this.username = username;
        this.password = password;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }
}
```

Figure 12: RegisterClass

Δώσαμε μερικά στοιχεία που πιθανόν να είχε ένας πελάτης, δημιουργήσαμε τον κατάλληλο constructor και γράψαμε τις απαραίτητες getters and setters. Το σύστημα μας διαθέτει μόνο εναν admin οπότε θα κάνουμε παρακάτω έλεγχο για το εαν τα στοιχεία που δώσαμε είναι αυτά που γνωρίζουμε για τον admin αλλιως θα ψάξουμε μέσα στο αρχείο των customers για να δούμε εαν πρόκειται για πελάτη( **Figure: 13**)

```

//checking user type
Log.f("STAVROULAKIS VASILIS THESIS 2018");
for (RegisterClass num : customers_list) {
    if (a1.equals(num.getUsername()) & a2.equals(num.getPassword())) {
        Log.i("YOU LOGGED AS CUSTOMER");
        //System.out.println("YOU LOGGED AS CUSTOMER");
        //For shopping basket's purposes we must know and pass customer's username
        CustomerMainPanel.customerMainPanelMethod(a1);
    }
}
if (a1.equals(admin_username) && a2.equals(admin_password)) {
    //There is an only one admin, so we dont need its username
    Log.i("YOU LOGGED AS ADMIN");
    //System.out.println("YOU LOGGED AS ADMIN");
    AdminMainPanel.adminMainPanelMethod();
}
file.close();

```

Figure 13: CheckUser (b)

Εδώ ελέγχουμε εαν στη λίστα που ανακατασκευάσαμε απο το αρχείο customers.txt υπάρχει πελάτης σύμφωνα με τα στοιχεία που έδωσε ο χρήστης. Εαν ναι τότε έκανε είσοδο ως πελάτης και μεταβαίνει στο κεντρικό μενού(πάνελ) του πελάτη.

Εαν όμως δώσαμε στοιχεία που είναι ίδια με αυτά του admin ( a1 είναι το username , ενώ a2 το password του admin). Εαν είναι ίδια, τότε μεταβαίνουμε στο κεντρικό μενού(πάνελ) του admin. Φυσικά πάντα κλείνουμε τα αρχεία για να μην υπάρξουν προβλήματα.

#### 4.2.1 ΕΙΣΟΔΟΣ ΩΣ ADMIN ΣΤΟ ΣΥΣΤΗΜΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Εαν κάνουμε είσοδο ως admin τότε θα ανοίξει το παρακάτω JFrame(**Figure: 14**)

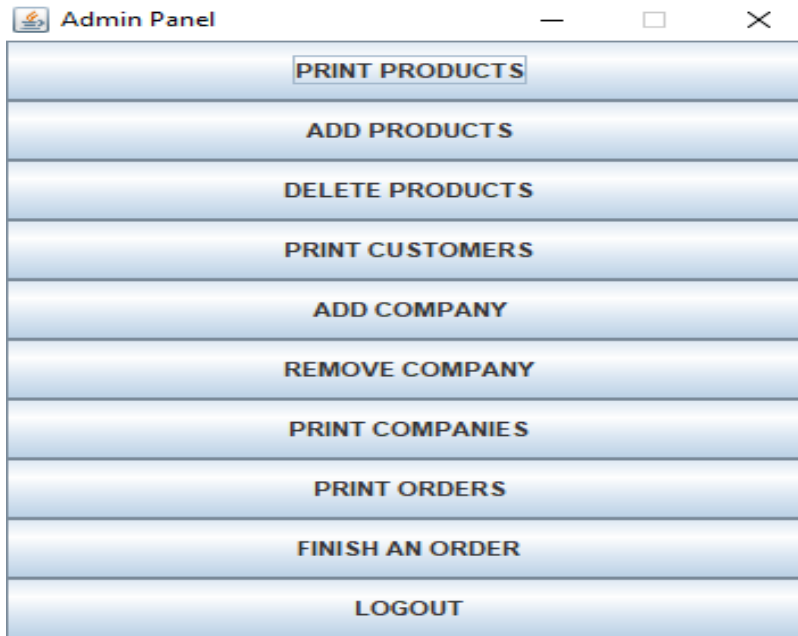


Figure 14: Admin Panel

Όπως βλέπουμε έχουμε υλοποιήσει τις παραπάνω λειτουργίες. Ένας admin δηλαδή μπορεί να εκτυπώσει τον κατάλογο προϊόντων, να προσθέσει προϊόντα στο υποκατάστημα να αφαιρέσει τα προϊόντα όπως και να εκτυπώσει πελάτες.

Επίσης μπορεί να προσθέσει/αφαιρέσει εταιρείες ή να οριστικοποιήσει μια παραγγελία που έχει κάνει πρωτύτερα ένας πελάτης. Τέλος μπορεί να δει τις παλιές παραγγελίες ενός πελάτη και να κάνει έξοδο από το σύστημα.

Αναλυτικά θα περιγράψουμε το πώς λειτουργεί από πλευράς κώδικα και το πώς εμφανίζεται η όλη λειτουργία στη συνέχεια. Αρχικά όμως ας δούμε το πώς δημιουργήθηκε το κεντρικό admin panel( **Figure : 15)**

```

public class AdminMainPanel {

    public static void adminMainPanelMethod() {
        JFrame frame = new JFrame("Admin Panel");
        frame.setSize(360, 400);
        frame.setLayout(new BorderLayout());
        JPanel panel = new JPanel();
        frame.add(panel);
        panel.setLayout(new GridLayout(10, 1));
        JButton btn1 = new JButton("PRINT PRODUCTS");
        JButton btn2 = new JButton("ADD PRODUCTS");
        JButton btn3 = new JButton("DELETE PRODUCTS");
        JButton btn4 = new JButton("PRINT CUSTOMERS");
        JButton btn5 = new JButton("ADD COMPANY");
        JButton btn6 = new JButton("REMOVE COMPANY");
        JButton btn7 = new JButton("PRINT COMPANIES");
        JButton btn8 = new JButton("PRINT ORDERS");
        JButton btn9 = new JButton("FINISH AN ORDER");
        JButton btn10 = new JButton("LOGOUT");

        panel.add(btn1);
        panel.add(btn2);
        panel.add(btn3);
        panel.add(btn4);
        panel.add(btn5);
        panel.add(btn6);
    }
}

```

Figure 15: admin main panel (a)

Παραπάνω φαίνονται οι λειτουργίες του admin. Δημιουργήσαμε JFrame και ένα panel και ενσωματώσαμε σε αυτό κάποια buttons που υλοποιούν τις παραπάνω λειτουργίες. Στη συνέχεια βλέπουμε ότι (**Figure: 16**) κάθε κουμπί μας στέλνει σε διαφορετική κλάση.

```

//actionListeners for buttons
btn1.addActionListener(new PrintProducts());
btn2.addActionListener(new AddProducts());
btn3.addActionListener(new DeleteProducts());
btn4.addActionListener(new PrintCustomers());
btn5.addActionListener(new AddCompany());
btn6.addActionListener(new RemoveCompany());
btn7.addActionListener(new PrintCompanies());
btn8.addActionListener(new PrintFinishedOrders());
btn9.addActionListener(new FinishCustomerOrder());
btn10.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        frame.dispose();
    }
});

frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
frame.setResizable(false);
}

```

Figure 16: admin main panel (b)

Θα περιγράψουμε κάθε κλάση ξεχωριστά στις παρακάτω υποενότητες.

#### 4.2.1.1 ADD PRODUCTS

Η κλάση AddProducts.java ουσιαστικά μας στέλνει με κατάλληλα κουμπιά σε διαφορετικά JFrame για να προσθέσουμε προϊόντα ανάλογα με το είδος τους. Για να προσθέσουμε μια κάμερα δημιουργούμε ένα JFrame και ένα JPanel. Αυτό φαίνεται παρακάτω(**Figure: 17**)

```
@Override
public void actionPerformed(ActionEvent ae) {
    JFrame frame = new JFrame("Add Camera");
    frame.setSize(250, 340);
    frame.setLayout(new BorderLayout());
    JPanel panel = new JPanel();
    frame.add(panel);
    panel.setLayout(new GridLayout(17, 1));
    JLabel label1 = new JLabel("Camera model");
    JLabel label2 = new JLabel("Camera monitor size");
    JLabel label3 = new JLabel("Camera megapixels");
    JLabel label4 = new JLabel("Camera frames/second");
    JLabel label5 = new JLabel("Camera recording type");
    JLabel label6 = new JLabel("Camera price");
    JLabel label7 = new JLabel("Store name");
    JLabel label8 = new JLabel("Camera quantity");
    JTextField field1 = new JTextField(25);
    JTextField field2 = new JTextField(25);
    JTextField field3 = new JTextField(25);
    JTextField field4 = new JTextField(25);
    JTextField field5 = new JTextField(25);
    JTextField field6 = new JTextField(25);
    JTextField field7 = new JTextField(25);
    JTextField field8 = new JTextField(25);
    --
}
```

Figure 17: Add Camera

Κατόπιν τα κατάλληλα JTextFields, JLabels με σκοπό μέσω ενός ActionListener να μπορέσουμε να δημιουργήσουμε ένα νέο αντικείμενο τύπου κάμερας. Επιπρόσθετα ο αριθμός των αντικειμένων που δημιουργούμε είναι σύμφωνα με την ποσότητα των καμερών που έδωσε ο πελάτης στη φόρμα του προϊόντος.

Κάθε φορά καλείται η κλάση writeProduct (εδώ writeCamera) η οποία καλείται τόσες φορές, όσες είναι τα αντικείμενα που δημιουργήσαμε. Η συνέχεια του κώδικα φαίνεται παρακάτω(**Figure: 18**)

```

btn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        String text1 = field1.getText();
        String text2 = field2.getText();
        String text3 = field3.getText();
        String text4 = field4.getText();
        String text5 = field5.getText();
        String text6 = field6.getText();
        String text7 = field7.getText();
        String text8 = field8.getText();
        //Add by using quantity given
        int quantity = Integer.parseInt(text8);
        for (int i = 0; i < quantity; i++) {
            CameraClass new_camera = new CameraClass(text1, text2, text3, text4, text5, text6, text7);
            writeCamera(new_camera);
            Log.i("YOU ADDED A CAMERA ITEM");
        }
        frame.dispose();
    }
});

```

Figure 18: Add camera (b)

Τέλος, η κλάση ανοίγει το αρχείο cameras.txt και γράφει τα αντικείμενα στο αρχείο. Κάθε αντικείμενο γράφεται με την βοήθεια της toString() κάθε κλάσης(προϊόντος) και τα κατάλληλα FileWriter και BufferedReader(**Figure: 19**)

```

public void writeCamera(CameraClass c) {
    try {
        String current = new java.io.File(".").getCanonicalPath();
        File file = new File(current + "\\cameras.txt");
        if (!file.exists()) {
            file.createNewFile();
        }
        FileWriter fw = new FileWriter(file, true);
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("\n" + c.toString());
        bw.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Figure 19: Add camera (c)

Το JFrame της προσθήκης μιας κάμερας φαίνεται παρακάτω(**Figure: 20**)

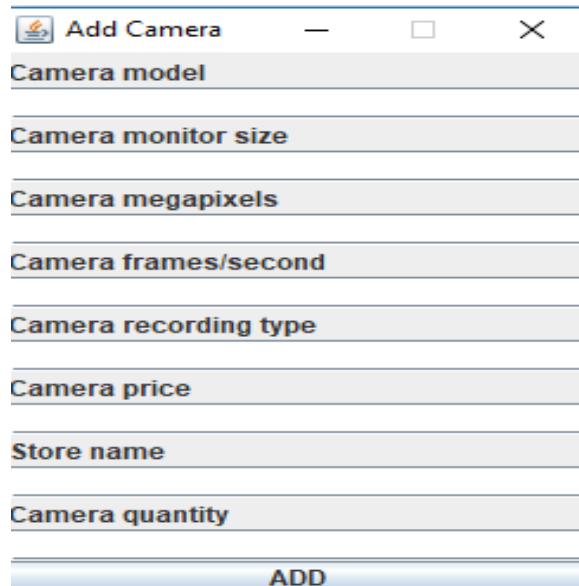
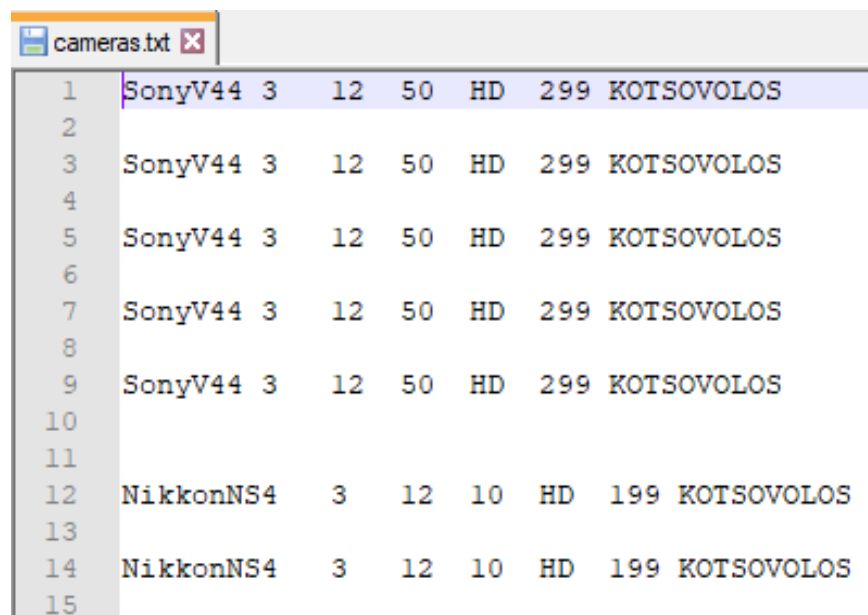


Figure 20: Add camera

Τώρα θα δούμε το πώς μπορούμε να προσθέσουμε για παράδειγμα 5 κάμερες στο αρχείο cameras.txt. Το αρχικό αρχείο ήδη είχε μερικές κάμερες μέσα(**Figure: 21**)



Line	Camera model	Quantity	Monitor size	Megapixels	Frames/second	Recording type	Price	Store name
1	SonyV44	3	12	50	HD	299	KOTSOVOLOS	
2								
3	SonyV44	3	12	50	HD	299	KOTSOVOLOS	
4								
5	SonyV44	3	12	50	HD	299	KOTSOVOLOS	
6								
7	SonyV44	3	12	50	HD	299	KOTSOVOLOS	
8								
9	SonyV44	3	12	50	HD	299	KOTSOVOLOS	
10								
11								
12	NikonNS4	3	12	10	HD	199	KOTSOVOLOS	
13								
14	NikonNS4	3	12	10	HD	199	KOTSOVOLOS	
15								

Figure 21: Αρχείο cameras.txt πριν την εισαγωγή νέων καμερών

Μετά την προσθήκη των παρακάτω στοιχείων στη φόρμα(**Figure: 22**)



<b>Camera model</b>	SonyTSVN983001
<b>Camera monitor size</b>	3
<b>Camera megapixels</b>	18
<b>Camera frames/second</b>	55
<b>Camera recording type</b>	FullHD
<b>Camera price</b>	4999
<b>Store name</b>	KOTSOVOLOS
<b>Camera quantity</b>	12
<b>ADD</b>	

Figure 22: Φορμα add camera

Μετά την υποβολή των παραπάνω στοιχείων το αρχείο έχει αλλάξει( **Figure: 23**)

Line	Camera Model	Monitor Size	Megapixels	Frames/Second	Recording Type	Price	Store Name
1	SonyV44	3	12	50	HD	299	KOTSOVOLOS
2							
3	SonyV44	3	12	50	HD	299	KOTSOVOLOS
4							
5	SonyV44	3	12	50	HD	299	KOTSOVOLOS
6							
7	SonyV44	3	12	50	HD	299	KOTSOVOLOS
8							
9	SonyV44	3	12	50	HD	299	KOTSOVOLOS
10							
11							
12	NikonNS4	3	12	10	HD	199	KOTSOVOLOS
13							
14	NikonNS4	3	12	10	HD	199	KOTSOVOLOS
15							
16	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
17							
18	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
19							
20	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
21							
22	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
23							
24	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
25							
26	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
27							
28	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
29							
30	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
31							
32	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
33							
34	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS
35							
36	SonyTSVN983001	3.0	18	55	FullHD	4999	KOTSOVOLOS

Figure 23: Αρχείο cameras.txt μετά την εισαγωγή (κομμάτι αυτού)

Με παρομοιο τρόπο λειτουργούν και οι υπόλοιπες προσθήκες των προϊόντων για παράδειγμα εαν θέλουμε να προσθέσουμε κονσόλες παιχνιδιών στο κατάστημα μας τότε το κατάλληλο JFrame ( φορμα προσθήκης προϊόντος ) που εμφανίζεται στον admin είναι το παρακάτω(**Figure: 24**)

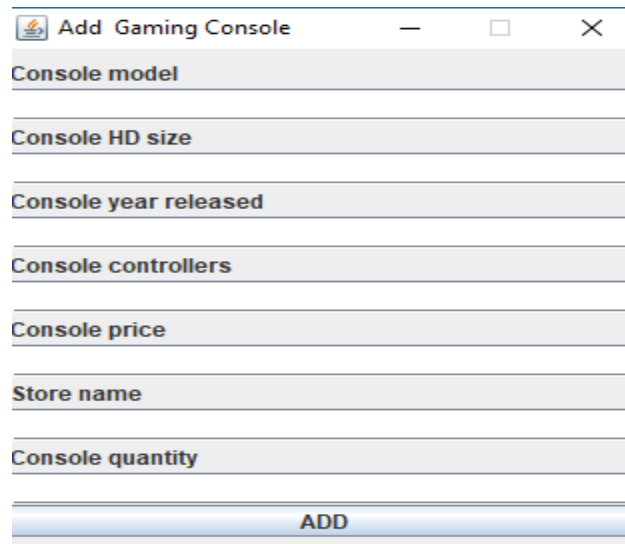
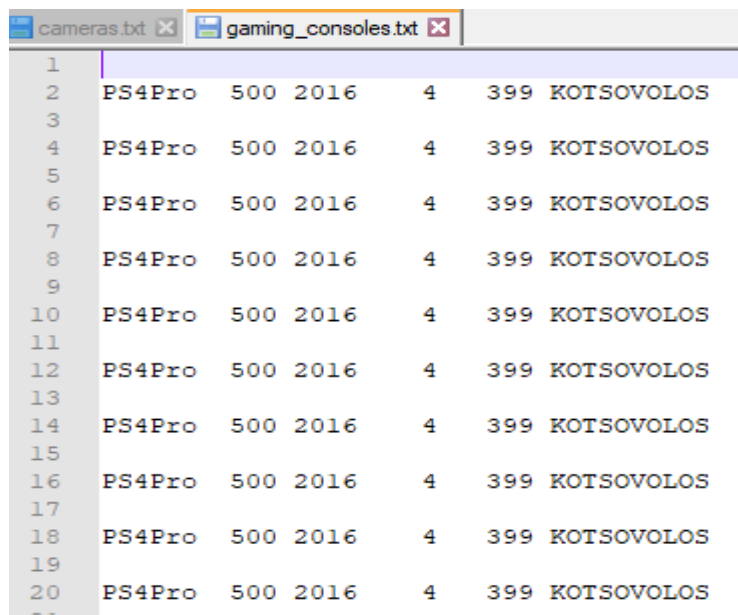


Figure 24: Φορμα προσθήκης gaming console

Ενώ το αρχείο προτού γίνει update με το νέο προϊόν/οντα είναι το εξής(**Figure: 25**)



Line	Console Model	HD Size	Year Released	Controllers	Price	Store Name
1						
2	PS4Pro	500	2016	4	399	KOTISOVOLOS
3						
4	PS4Pro	500	2016	4	399	KOTISOVOLOS
5						
6	PS4Pro	500	2016	4	399	KOTISOVOLOS
7						
8	PS4Pro	500	2016	4	399	KOTISOVOLOS
9						
10	PS4Pro	500	2016	4	399	KOTISOVOLOS
11						
12	PS4Pro	500	2016	4	399	KOTISOVOLOS
13						
14	PS4Pro	500	2016	4	399	KOTISOVOLOS
15						
16	PS4Pro	500	2016	4	399	KOTISOVOLOS
17						
18	PS4Pro	500	2016	4	399	KOTISOVOLOS
19						
20	PS4Pro	500	2016	4	399	KOTISOVOLOS
21						

Figure 25: Αρχείο gaming\_consoles πριν την εισαγωγή κονσόλας

Εαν προσθέσουμε το παρακάτω προϊόν(**Figure: 26**)

The screenshot shows a window titled "Add Gaming Console" with the following fields and values:

Console model	NintendoWII
Console HD size	500
Console year released	2015
Console controllers	6
Console price	599
Store name	KOTSOVOLOS
Console quantity	3

At the bottom of the form is an "ADD" button.

Figure 26: Η προσθήκη μιας κονσόλας στο αρχείο

Τότε το αρχείο gaming\_consoles.txt θα αλλάξει( **Figure: 27**)

The screenshot shows a text editor with two tabs: "cameras.txt" and "gaming\_consoles.txt". The "gaming\_consoles.txt" tab is active and displays the following content:

```
1  
2 PS4Pro 500 2016 4 399 KOTSOVOLOS  
3  
4 PS4Pro 500 2016 4 399 KOTSOVOLOS  
5  
6 PS4Pro 500 2016 4 399 KOTSOVOLOS  
7  
8 PS4Pro 500 2016 4 399 KOTSOVOLOS  
9  
10 PS4Pro 500 2016 4 399 KOTSOVOLOS  
11  
12 PS4Pro 500 2016 4 399 KOTSOVOLOS  
13  
14 PS4Pro 500 2016 4 399 KOTSOVOLOS  
15  
16 PS4Pro 500 2016 4 399 KOTSOVOLOS  
17  
18 PS4Pro 500 2016 4 399 KOTSOVOLOS  
19  
20 PS4Pro 500 2016 4 399 KOTSOVOLOS  
21  
22 NintendoWII 500 2015 6 599 KOTSOVOLOS  
23  
24 NintendoWII 500 2015 6 599 KOTSOVOLOS  
25  
26 NintendoWII 500 2015 6 599 KOTSOVOLOS
```

Figure 27: Το αρχείο gaming\_consoles μετά την προσθήκη κονσόλων

#### 4.2.1.2 PRINT PRODUCTS

Επιλέξαμε να κάνουμε εκτύπωση των συνολικών προϊόντων. Μιας και έχουμε 5 προϊόντα στο κατάστημα, ουσιαστικά θα ανοίξουμε τα 3 αρχεία, κατόπιν θα τα φορτώσουμε σε κατάλληλα `arrayLists` και θα τα κάνουμε εκτύπωση χρησιμοποιώντας `JTextArea` και `JScrollPane`. Ο κώδικας για να φορτώσουμε αρχεία φαίνεται παρακάτω( **Figure: 28**)

```

while (file1.hasNext()) {
    String line = file1.nextLine();
    if (!line.isEmpty()) {
        String[] stringArr = line.split("\t");
        String cameraModel = stringArr[0];
        String cameraMonitorSize = stringArr[1];
        String cameraMP = stringArr[2];
        String cameraFramesPerSecond = stringArr[3];
        String cameraRecordingType = stringArr[4];
        String cameraPrice = stringArr[5];
        String cameraStoreName = stringArr[6];
        CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cam
        cameras_list.add(camera);
    }
}

while (file2.hasNext()) {
    String line = file2.nextLine();
    if (!line.isEmpty()) {
        String[] stringArr = line.split("\t");
        String consoleModel = stringArr[0];
        String consoleSize = stringArr[1];
        String consoleYearReleased = stringArr[2];
        String consoleControlllers = stringArr[3];
        String consolePrice = stringArr[4];
        String consoleStoreName = stringArr[5];
        GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, conso
        gamingConsoles_list.add(console);
    }
}

while (file3.hasNext()) {

```

Figure 28: Φόρτωση αρχείων για εκτύπωση(κομμάτι κώδικα)

Δημιουργούμε JFrame και JTextArea που θα ενσωματωθεί στο frame. Στη συνέχεια το JScrollPane θα ενσωματωθεί στο frame ενώ κάθε φορά θα διαβάζουμε τα αντικείμενα από κάθε αρχείο και με κατάλληλη μορφοποίηση θα εμφανιστούν στην οθόνη( **Figure: 29**)

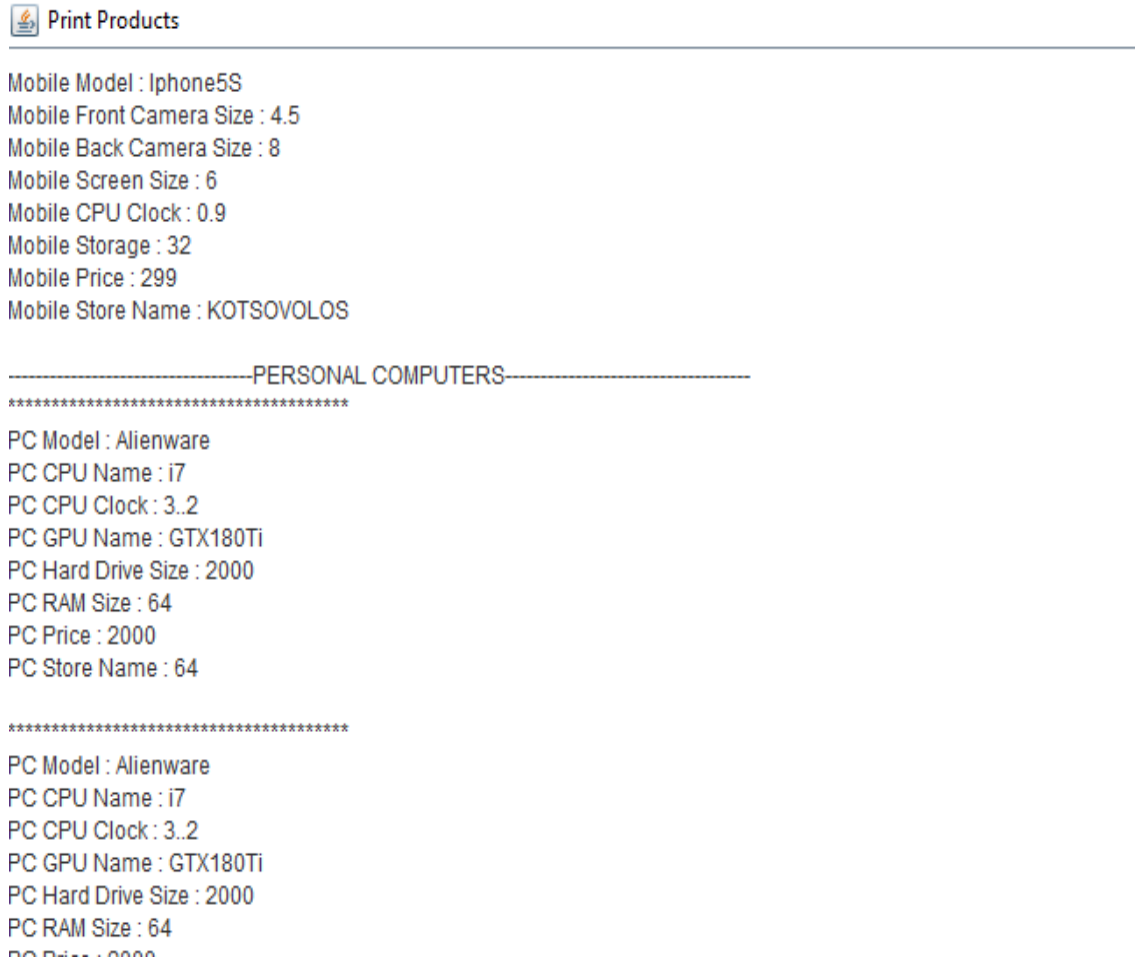


Figure 29: Εκτύπωση συνολικών προϊόντων

Ενώ ο κώδικας φαίνεται παρακάτω(**Figure:30**)

```

for (GamingConsoleClass gm : gamingConsoles_list) {
    textArea.append("*****" + "\n");
    textArea.append("Console Model : " + gm.getConsole_name() + "\n");
    textArea.append("Console HD Size : " + gm.getConsole_size() + "\n");
    textArea.append("Console Controllers : " + gm.getConsole_controllers_count() + "\n");
    textArea.append("Console Price : " + gm.getConsole_price() + "\n");
    textArea.append("Console Store Name : " + gm.getStore_name() + "\n");
    textArea.append("\n");
}
textArea.append("-----MOBILES-----");
textArea.append("\n");
for (MobileClass m : mobiles_list) {
    textArea.append("*****" + "\n");
    textArea.append("Mobile Model : " + m.getMobile_model() + "\n");
    textArea.append("Mobile Front Camera Size : " + m.getFront_camera_mpixel() + "\n");
    textArea.append("Mobile Back Camera Size : " + m.getBack_camera_mpixel() + "\n");
    textArea.append("Mobile Screen Size : " + m.getMobile_screen_size() + "\n");
    textArea.append("Mobile CPU Clock : " + m.getMobile_cpu() + "\n");
    textArea.append("Mobile Storage : " + m.getMobile_storage() + "\n");
    textArea.append("Mobile Price : " + m.getMobile_price() + "\n");
    textArea.append("Mobile Store Name : " + m.getStore_name() + "\n");
    textArea.append("\n");
}

textArea.append("-----PERSONAL COMPUTERS-----");
textArea.append("\n");
for (PersonalComputerClass pc : personalComputers_list) {
    textArea.append("*****" + "\n");
    textArea.append("PC Model : " + pc.getModel_name() + "\n");
    textArea.append("PC CPU Name : " + pc.getPc_cpu_name() + "\n");

```

Figure 30: Εκτύπωση προϊόντων (κομμάτι κώδικα)

#### 4.2.1.3 DELETE PRODUCTS

Για να σβήσουμε ένα προϊόν πρέπει καταρχάς να ξέρουμε κάποια χαρακτηριστικά για το συγκεκριμένο προϊόν, όπως είναι για παράδειγμα το είδος του. Έτσι, δημιουργήσαμε την κλάση DeleteProducts.java με σκοπό να μπορούμε να μεταβούμε στην κατάλληλη διαγραφή προϊόντος. Μέρος του κώδικα της κλάσης φαίνεται παρακάτω(**Figure: 31**)

```

frame.add(panel);
panel.setLayout(new GridLayout(6, 1));
JButton btn1 = new JButton("REMOVE CAMERAS");
JButton btn2 = new JButton("REMOVE TELEVISIONS");
JButton btn3 = new JButton("REMOVE MOBILES");
JButton btn4 = new JButton("REMOVE PERSONAL COMPUTERS"); //
JButton btn5 = new JButton("REMOVE GAMING CONSOLES");
JButton btn6 = new JButton("BACK");

panel.add(btn1);
panel.add(btn2);
panel.add(btn3);
panel.add(btn4);
panel.add(btn5);
panel.add(btn6);

//actionListeners for buttons
btn1.addActionListener(new RemoveCamera());
btn2.addActionListener(new RemoveTV());
btn3.addActionListener(new RemoveMobiles());
btn4.addActionListener(new RemovePC());
btn5.addActionListener(new RemoveGamingConsoles());
btn6.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        frame.dispose();
    }
});

```

Figure 31: Κώδικας μετάβασης για διαγραφή κατάλληλου προϊόντος

Ο κώδικας είναι παρόμοιος σε κάθε περίπτωση, οπότε θα εξηγήσουμε μία από τις παραπάνω διαγραφές προϊόντων. Για παράδειγμα έστω ότι θέλουμε να διαγράψουμε έναν ηλεκτρονικό υπολογιστή από το αρχείο personalComputers.txt. Αρχικά μεταβαίνουμε στη κλάση πατώντας το κατάλληλο κουμπί από το frame της κλάσης DeleteProducts.java( **Figure: 32**)

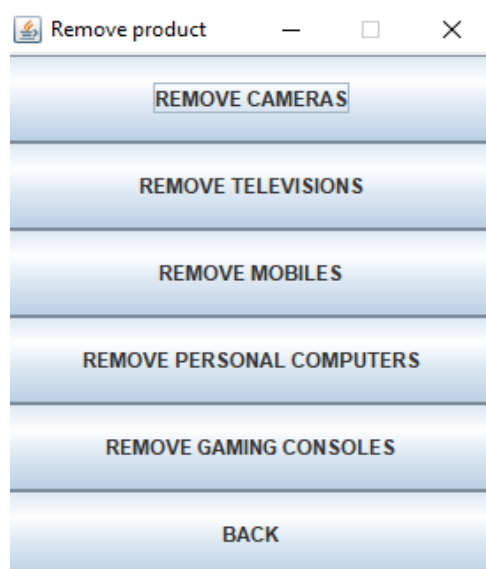


Figure 32: Διαγραφή προϊόντος(κεντρικό πανελ)

Αρχικά θα μεταβούμε στην κλάση RemovePC. Θα δημιουργήσουμε ένα κατάλληλο JFrame και θα δώσουμε τα κατάλληλα στοιχεία για την διαγραφή(**Figure: 33**)

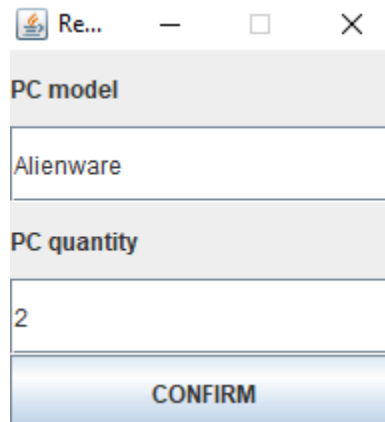


Figure 33: Διαγραφή ενός υπολογιστή από το αρχείο

Ουσιαστικά μέσω του confirm ο ActionListener παίρνει τις παραπάνω τιμές και τις στέλνει στη κλάση reconstructArrayList. Μέσω αυτής της κλάσης, ανακατασκευάζουμε το arrayList με τους υπολογιστές από το αρχείο(**Figure: 34**)

```

Scanner file = null;
ArrayList<PersonalComputerClass> pc_list = new ArrayList<>();
try {
    file = new Scanner(new File("personalComputers.txt"));
} catch (FileNotFoundException ex) {
    Logger.getLogger(RemovePC.class.getName()).log(Level.SEVERE, null, ex);
}
while (file.hasNext()) {
    String line = file.nextLine();
    if (!line.isEmpty()) {
        String[] stringArr = line.split("\t");
        String pcName = stringArr[0];
        String pcCpuName = stringArr[1];
        String cpuClock = stringArr[2];
        String gpuName = stringArr[3];
        String pcHardDrive = stringArr[4];
        String pcRamSize = stringArr[5];
        String pcPrice = stringArr[6];
        String pcStoreName = stringArr[7];
        PersonalComputerClass pc = new PersonalComputerClass(pcName, pcCpuName, cpuClock, gpuName, pcHardDrive,
pc_list.add(pc);
    }
}

```

Figure 34: Ανακατασκευή του arrayList pcs\_list



Κατόπιν χρησιμοποιώντας μια κλάση που καλείται μέσα στη προηγούμενη κλάση που ονομάζεται `removeUsingIterator` προσπαθούμε, έχοντας το μοντέλο αλλά και την ποσότητα(τα έδωσε ο admin στην φόρμα) έχοντας ως μέγιστο όριο τον αριθμό/ποσότητα που δώσαμε, να προσπαθεί μέσω ενός iterator κάθε φορά να αφαιρεί από τη `arrayList` ένα προϊόν που έχει όνομα μοντέλου ίδιο με αυτό που έδωσε ο admin.

Όταν ο αριθμός προϊόντων που δώσαμε ισούται με τον αριθμό που κλήθηκε και βρήκε/διέγραψε η κλάση, τότε η διαγραφή σταματά και βγαίνει ως έξοδο στο console, ένα μήνυμα επιτυχίας(**Figure: 35**)

```

int count_times = pcQuantity_integer, product_count = 0;
product_count = removeUsingIterator(count_times, pcQuantity_integer, pc_list, pcModelName);
System.out.println(" YOU SUCCESSFULLY REMOVED:" + product_count + " ITEMS!");
if (product_count == 0) {
    Log.w(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");
    //System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");
}
saveListToFile(pc_list, "personalComputers.txt");
}

public int removeUsingIterator(int count_times, int pcQuantity_integer, ArrayList pc_list, String pcModelName)
    Iterator<PersonalComputerClass> it = pc_list.iterator();
    int product_count = 0;
    while (it.hasNext()) {
        if (it.next().getModel_name().equals(pcModelName)) {
            count_times--;
            it.remove();
            product_count++;
        }
        if (count_times == 0) {
            break;
        }
    }
}

```

Figure 35: Διαγραφή ποσότητας προϊόντων από `arrayList`

Τέλος πρέπει να αποθηκευτεί η νέα και ανανεωμένη `arrayList` των ηλεκτρονικών υπολογιστών στο αρχείο με τη χρήση της κλάσης `saveListToFile`(**Figure:36**)

```

public void saveListToFile(ArrayList pc_list, String fileName) throws FileNotFoundException {
    PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));
    for (Object pc : pc_list) {
        pw.println(pc.toString());
    }
    pw.close();
}

```

Figure 36: Αποθήκευση της νέας arrayList στο αρχείο

Ομως ας δούμε μια διαγραφή(το πώς ήταν και πως αλλαξε το αρχείο). Αρχικά το αρχείο περιείχε τα εξής προϊόντα(**Figure: 37**)

Line	Brand	Model	OS	Processor	RAM	Storage	Price	Manufacturer
1	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
2								
3	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
4								
5	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
6								
7	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
8								
9	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
10								
11								

Figure 37: Αρχείο personalComputers πριν τη διαγραφή

Τελικά το αρχείο αλλαξε(**Figure:38**)

Line	Brand	Model	OS	Processor	RAM	Storage	Price	Manufacturer
1	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
2								
3	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
4								
5	Alienware	i7	3..2	GTX180Ti	2000	64	2850	KOTSOVOLOS
6								
7								

Figure 38: Αρχείο personalComputers μετα την διαγραφή

#### 4.2.1.4 PRINT CUSTOMERS

Με παρόμοιο τρόπο που εκτυπώσαμε τα προϊόντα μπορούμε απλά να εκτυπώσουμε τους πελάτες του καταστήματος. Αρχικά φορτώση του αρχείου των πελατών, ανακατασκευή της arrayList και στη συνέχεια εκτύπωση με JTextArea και JScrollPane(**Figure: 39**)

```

        String city = stringArr[3];
        String TK = stringArr[4];
        String AFM = stringArr[5];
        String username = stringArr[6];
        String password = stringArr[7];

        RegisterClass customers = new RegisterClass(fname, surname, address, city, TK, AFM, use
        customers_list.add(customers);
    }

}

JFrame frame = new JFrame("Print Customers");
frame.setLayout(new BorderLayout());
frame.setSize(800, 600);
JTextArea textArea = new JTextArea();
frame.add(textArea);
textArea.setEditable(false);
frame.getContentPane().add(textArea);
for (RegisterClass s : customers_list) {
    textArea.append("*****" + "\n");
    textArea.append("Firstname : " + s.getFname() + "\n");
    textArea.append("Lastname : " + s.getSurname() + "\n");
    textArea.append("Address : " + s.getAddress() + "\n");
    textArea.append("City : " + s.getCity() + "\n");
    textArea.append("TK : " + s.getTK() + "\n");
    textArea.append("AFM : " + s.getAFM() + "\n");
    textArea.append("Username : " + s.getUsername() + "\n");
    textArea.append("Password : " + s.getPassword() + "\n");
    textArea.append("\n");
}

```

Figure 39: Εκτύπωση πελατών(κομμάτι κώδικα)

Όσο αφορά την εκτύπωση, πατώντας το κουμπί PRINT CUSTOMERS που βρίσκεται στο κεντρικό πανελ του admin μπορούμε να δούμε τους πελάτες του καταστήματος(**Figure: 40**)

```
Print Customers
*****
Firstname : Ioannis
Lastname : Mixelakis
Address : Athitaki 72
City : Heraklion
TK : 71410
AFM : 78728987
Username : mike44
Password : 827283
*****
Firstname : Maria
Lastname : Epimenidou
Address : Athinagora 12
City : Chania
TK : 7483
AFM : 78938928
Username : maria32
Password : 827827
*****
Firstname : Sofia
Lastname : Lamioni
Address : Polykastrou 548
City : Attiki
TK : 2829
AFM : 839872983
Username : sofia1
Password : 8282933
```

Figure 40: Εκτύπωση πελατών

#### 4.2.1.5 ADD COMPANY

Η προσθήκη μιας εταιρείας γίνεται όπως ακριβώς η προσθήκη προϊόντων. Απλά αντί να ανοίξουμε 5 αρχεία, ανοίγουμε μόνο το αρχείο των εταιρειών ανακατασκευάζουμε το arrayList των εταιρειών, προσθέτουμε μια εταιρεία, και κάνουμε με κατάλληλη κλάση(save) στο αρχείο. Πιο συγκεκριμένα, το JFrame προσθήκης μιας εταιρείας παρουσιάζεται παρακάτω(**Figure: 41**)



Figure 41: JFrame προσθήκης εταιρείας στο κατάστημα

Αρχικά ανακατασκευή της arrayList των εταιρειών(**Figure: 42**)

```

while (file.hasNext()) {
    String line = file.nextLine();
    if (!line.isEmpty()) {
        String[] stringArr = line.split("\t");
        String name = stringArr[0];
        String owner = stringArr[1];
        String ITN = stringArr[2];
        String location = stringArr[3];
        CompanyClass companies = new CompanyClass(name, owner, ITN, location);
        companies_list.add(companies);
    }
}

```

Figure 42: Ανακατασκευή της arrayList των εταιρειων

Ενα σημείο που πρέπει να προσέξουμε εδώ είναι ότι **ελέγχουμε για το εάν υπάρχει ήδη μια εταιρεία στο σύστημα μας** με τον ίδιο αριθμό φορολογικού μητρώπου (I.T.N). Αυτό γίνεται με το εάν μετρήσουμε πόσες εταιρείες υπάρχουν με ίδιο ITN σύμφωνα με τον αριθμό ITN που δώσαμε για την προσθήκη νέας εταιρείας.

Εάν δεν υπάρχει άλλη εταιρεία τότε καλούμε τη κλάση writeCompany αλλιώς βγάζουμε ένα μήνυμα προειδοποίησης και δεν γίνεται προσθήκη της εταιρείας (**Figure: 43**)

```

int j = 0;
for (int i = 0; i < companies_list.size(); i++) {
    CompanyClass cc = companies_list.get(i);
    if (cc.getCompany_ITN().equals(text3)) {
        j = i + 1;
    }
}
if (j <= 0) {
    CompanyClass new_company = new CompanyClass(text1, text2, text3, text4);
    writeCompany(new_company);
}
if (j > 0) {
    Log.i("THIS COMPANY IS IN OUR SYSTEM!");
}

```

Figure 43: Έλεγχος για ύπαρξη όμοιας εταιρείας στο σύστημα

Τελος γράφουμε με τον παραδοσιακό τρόπο στο σύστημα μας την εταιρεία(**Figure:44**)

```
public void writeCompany(CompanyClass c) {  
    try {  
        String current = new java.io.File(".").getCanonicalPath();  
        File file = new File(current + "\\companies.txt");  
        if (!file.exists()) {  
            file.createNewFile();  
        }  
        FileWriter fw = new FileWriter(file, true);  
        BufferedWriter bw = new BufferedWriter(fw);  
        bw.write("\n" + c.toString());  
        bw.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Figure 44: Προσθήκη εταιρείας στο σύστημα

Τώρα θα προσθέσουμε μια εταιρεία στο σύστημα. Το αρχικο αρχείο έχει την εξής μορφή(**Figure: 45**)

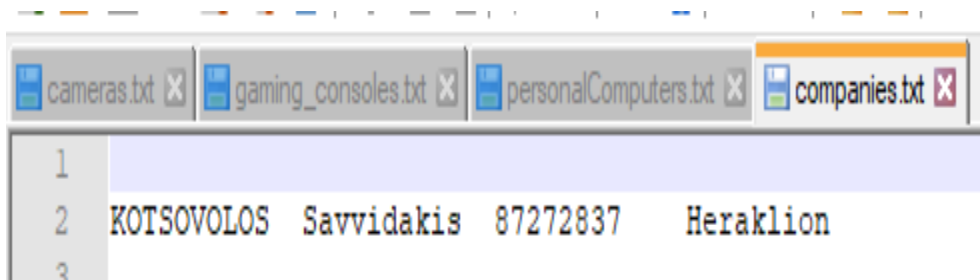
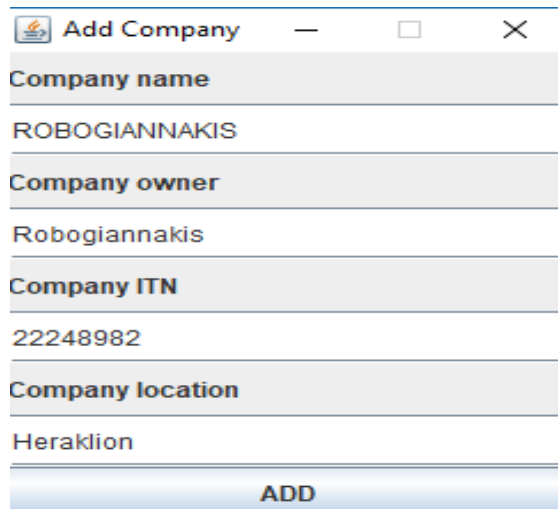


Figure 45: Αρχείο εταιρειών πριν την προσθήκη νεας εταιρείας

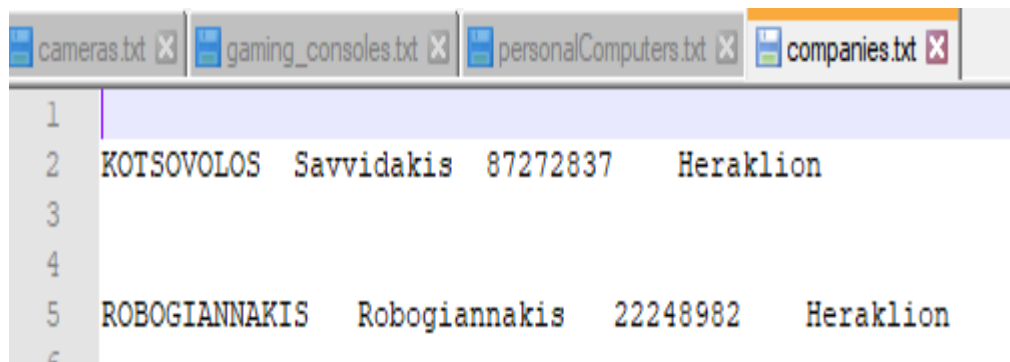
Εαν προσθέσουμε την παρακάτω εταιρεία(**Figure: 46**)



<b>Company name</b>
ROBOGIANNAKIS
<b>Company owner</b>
Robogiannakis
<b>Company ITN</b>
22248982
<b>Company location</b>
Heraklion
<b>ADD</b>

Figure 46: Προσθήκη εταιρείας στο αρχείο

Το αρχείο θα αλλάξει(**Figure: 47**)



```
1  
2 KOTSOVOLOS Savvidakis 87272837 Heraklion  
3  
4  
5 ROBOGIANNAKIS Robogiannakis 22248982 Heraklion  
6
```

Figure 47: Αρχείο μετά την προσθήκη εταιρείας

#### 4.2.1.6 REMOVE COMPANY & PRINT COMPANIES

Παρομοια με τις προηγούμενες διαγραφές, γίνεται και για την διαγραφή μιας εταιρείας. Όπως πάντα πρέπει να ανακατασκευάσουμε τη λίστα και στη συνέχεια με τη χρήση του ονόματος της να την αφαιρέσουμε από τη λίστα, και στη συνέχεια να αποθηκεύσουμε ξανά το αρχείο.

Η εκτύπωση είναι ακριβώς ίδια όπως με την εκτύπωση των προϊόντων και των πελατών αλλά αλλάξαμε τα αρχεία αλλά και τους τρόπους ανακατασκευάσματος των ArrayLists που στη συνέχεια ως γνωστόν θα εκτυπωθούν σε ένα JTextArea με χρήση JScrollPane

#### 4.2.1.7 FINISH CUSTOMER ORDER

Ο κώδικας εδώ είναι **αρκετά πιο περίπλοκος** και αξίζει να τον περιγράψουμε όσο πιο αναλυτικά γίνεται. Αρχικά δημιουργούμε ένα JFrame , JPanel αλλά και ένα κουμπί το CONFIRM. Επίσης ένα JTextField(με το κατάλληλο JLabel) το οποίο θα περιέχει το username του πελάτη. Έστω ότι ο πελάτης έχει κάνει μια παραγγελία.

Ο ActionListener θα πάρει την τιμή του username και θα το περάσει στη κλάση **reconstructTemporaryAndNoTemporaryFilesArrayListsPlusCalculateOrderCost** η οποία σκοπό έχει να κάνει ανακατασκευασμό των temporary files.

Τα temporary μας βοηθούν ως καλάθι αγορών του πελάτη καθώς όταν ο πελάτης κάνει μια παραγγελία τα προϊόντα γράφονται στα εκάστοτε temporary αρχεία προϊόντων τα οποία φυσικά αδειάζουν όταν ο admin οριστικοποιήσει την παραγγελία ώστε ο πελάτης μελλοντικά να μπορέσει να ξανα-αγοράσει προϊόντα.

Η παραπάνω κλάση αφού κάνει ότι πρέπει, καλεί την κλάση **calculateOrderCost** με σκοπό να υπολογίσει το τελικό κόστος των παραγγελιών και να το επιστρέψει στο σημείο που έγινε το event του ActionListener. Πολύ απλά, θεωρούμε ότι το αρχικό κόστος της παραγγελίας είναι μηδέν αλλά κάθε φορά που βλέπουμε σε κάθε αρχείο ένα προϊόν με το συγκεκριμένο username το προσθέτουμε στο συνολικό κόστος. Κατόπιν καλούμε τη κλάση **reconstructBasicProductFiles**.

Μεσα σε αυτήν καλούμε μια κλάση που θα διαγράψει τα προϊόντα από τα βασικά αρχεία καλώντας διαδοχικά τις συναρτήσεις:

- deleteCamerasFromNonTemporary
- deleteConsolesFromNonTemporary
- deleteMobilesFromNonTemporary
- deletePCSFFromNonTemporary
- deleteTVSFFromNonTemporary

Σε κάθε μια από αυτές τις συναρτήσεις καλούνται με την σειρά οι συναρτήσεις αποθήκευσης των νέων ArrayLists στα αρχεία(βασικά αρχεία που περιέχουν τα προϊόντα). Δηλαδή θέλουμε εκτός από το να σβηστούν τα προϊόντα (στη συνέχεια θα εξηγηθεί) από το καλάθι αγορών αλλά να σβηστούν και από το κατάστημα με σκοπό ο admin να προσθέσει μελλοντικά νέα προϊόντα:

- saveCamerasListToFile



- saveConsolesListToFile
- saveMobilesListToFile
- savePCSLISTToFile
- saveTVSLISTToFile

**Η διαγραφή απο τα temporary files:** Όταν όλες οι παραπάνω συναρτήσεις που περιγράψαμε τελειώσουν την εργασία τους, γυρνάμε πίσω στο event του actionListener και έχοντας το username του πελάτη απλά αδειάζουμε τα αρχεία χρησιμοποιώντας το **file.delete()**; και **file.createNewFile()**; Όπου file είναι το όνομα αρχείου που θέλουμε να διαγράψουμε και να δημιουργήσουμε ξανά(κενό αρχείο).

Για να μπορέσουμε να εκτυπώσουμε μια παραγγελία πρέπει αρχικά πρώτα να την οριστικοποιήσουμε. Οποτε δημιουργήσαμε τη κλάση **createAnOrder** που παίρνει ως ορίσματα το username και το τελικό κόστος που υπολογίσαμε με την κατάλληλη αλληλουχία συναρτήσεων ανοίγουμε το customers.txt για να πάρουμε τα βασικά στοιχεία του πελάτη που έχει το συγκεκριμένο username( είναι μοναδικό για κάθε χρήστη) και στη συνέχεια δημιουργούμε μια arrayList με τα στοιχεία του πελάτη και το κόστος και τα αποθηκεύουμε στο αρχείο customer\_orders.txt(η αποθήκευση γίνεται με την βοήθεια της κλάσης **writeCustomerOrder**)

Για περισσότερη αληθοφάνεια δημιουργήσαμε τη κλάση **getTimeZone** που απλά βασίζομενοι σε μια γεννήτρια τυχαίων αριθμών επιλέγουμε ως ένα string μια ζώνη ώρας

Φυσικά κάθε παραγγελία έχει **διαφορετικό** order\_id που επιλέγεται και αυτό **τυχαία**. Ο κώδικας είναι αρκετά μεγάλος για να μπορέσει να παρουσιαστεί με εικόνες(κοιτάξτε το αρχείο **FinishCustomerOrder.java**)

#### 4.2.1.8 PRINT FINISHED ORDERS

Η εκτύπωση παραγγελιών είναι παρόμοια με την εκτύπωση προϊόντων, πελατών αλλά και εταιρειών. Άνοιγμα του αρχείου και στη συνέχεια ανακτασκή της λίστας και εκτύπωση με χρήση JTextArea και JScrollPane. Μέρος του κώδικα φαίνεται παρακάτω(**Figure: 48**)

```

//PRINTING
JFrame frame = new JFrame("Print Orders");
frame.setLayout(new BorderLayout());
frame.setSize(800, 500);
JTextArea textArea = new JTextArea();
frame.add(textArea);
JScrollPane sp = new JScrollPane(textArea);
textArea.setEditable(false);
JScrollPane scroll = new JScrollPane(textArea,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
frame.add(scroll);
textArea.append("-----ORDERS-----");
textArea.append("\n");
for (CustomerOrderClass c : finished_orders) {
    textArea.append("*****" + "\n");
    textArea.append("Customer Username : " + c.getUsername() + "\n");
    textArea.append("Customer : " + c.getSurname() + "\n");
    textArea.append("Customer Address : " + c.getAddress() + "\n");
    textArea.append("Customer City : " + c.getCity() + "\n");
    textArea.append("Customer Order ID : " + c.getOrder_id() + "\n");
    textArea.append("Customer TimeZone : " + c.getTime_zone() + "\n");
    textArea.append("Customer Finished(Days) : " + c.getTimeFinished() + "\n");
    textArea.append("Customer Order Cost : " + c.getOrder_cost() + "\n");
    textArea.append("\n");
}

```

Figure 48: Εκτύπωση παραγγελιών

#### 4.2.1.9 LOGOUT

Έξοδος απο το σύστημα κανοντας αυτόματα dispose το frame της κλάσης

### 4.3 Η ΠΛΕΥΡΑ ΤΟΥ ΠΕΛΑΤΗ ( CLIENT SIDE )

#### 4.3.1 PRINT PRODUCTS

Για να εκτυπώσουμε προϊόντα σε αυτή τη περίπτωση επιλέξαμε είτε να εκτυπωθούν συνολικά σε ένα JTextArea είτε να εκτυπωθούν ανα είδος. Ουσιαστικά στον πελάτη εμφανίζεται το εξής frame εκτυπώσεων(**Figure: 49**)

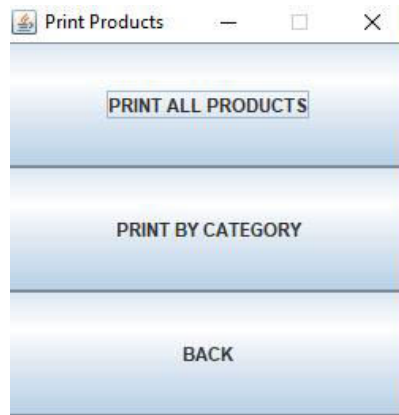


Figure 49: Δυο τρόποι εκτυπώσεων για τον πελάτη

Απλά δημιουργήσαμε ένα JFrame μέσα στο οποίο αφού προσθέσαμε τα κατάλληλα JButtons (Figure: 50)

```
//actionListeners for buttons also we know customer's username which is customer_username (
btn1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        JFrame frameForPrint = new JFrame("Print Products");
        frameForPrint.setSize(260, 270);
        frameForPrint.setLayout(new BorderLayout());
        JPanel panelForPrint = new JPanel();
        frameForPrint.add(panelForPrint);
        panelForPrint.setLayout(new GridLayout(3, 1));
        JButton btn1 = new JButton("PRINT ALL PRODUCTS");
        JButton btn2 = new JButton("PRINT BY CATEGORY");
        JButton btn3 = new JButton("BACK");

        panelForPrint.add(btn1);
        panelForPrint.add(btn2);
        panelForPrint.add(btn3);

        //actionListeners for buttons
        btn1.addActionListener(new PrintProducts());
        btn2.addActionListener(new PrintProductsByCategory());
        btn3.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                frameForPrint.dispose();
            }
        });
    }
});
```

Figure 50: Κώδικας υλοποίησης JFrame εκτυπώσεων

Η κλάση PrintProducts ανοίγει όλα τα αρχεία, ανακατασκευάζει τις κατάλληλες arrayList και μέσω της χρήσης JTextArea και JScrollPane εκτυπώνει στην οθόνη. Όπως δηλαδή κάναμε και στις εκτυπώσεις όλων των προϊόντων στον admin. Όμως ο πελάτης πιθανόν να θέλει να δει τα προϊόντα ξεχωριστά ανα

είδος. Οποτε μόλις μεταβούμε στη κλάση  
frame(**Figure: 51**).

PrintProductsByCategory μας ανοίγει το παρακάτω



Figure 51: Εκτυπωση προϊόντων ανα ειδος

Ο κώδικας φαίνεται παρακάτω(**Figure: 52**)

```
public void actionPerformed(ActionEvent ae) {
    JFrame frame = new JFrame("Print by Type");
    frame.setSize(360, 300);
    frame.setLayout(new BorderLayout());
    JPanel panel = new JPanel();
    frame.add(panel);
    panel.setLayout(new GridLayout(6, 1));
    JButton btn1 = new JButton("CAMERAS");
    JButton btn2 = new JButton("GAMING CONSOLES");
    JButton btn3 = new JButton("MOBILES");
    JButton btn4 = new JButton("PERSONAL COMPUTERS");
    JButton btn5 = new JButton("TVS");
    JButton btn6 = new JButton("BACK");
    panel.add(btn1);
    panel.add(btn2);
    panel.add(btn3);
    panel.add(btn4);
    panel.add(btn5);
    panel.add(btn6);

    //actionListeners for buttons
    btn1.addActionListener(new printCameras());
    btn2.addActionListener(new printGamingConsoles());
    btn3.addActionListener(new printMobiles());
    btn4.addActionListener(new printPersonalComputers());
    btn5.addActionListener(new printTVS());
    btn6.addActionListener(new ActionListener() {
        ~
        ~
    });
}
```

Figure 52: Κλήση συναρτήσεων για εκτυπωση διαφορετικών προϊόντων

Για παράδειγμα εαν θέλουμε να εκτυπώσουμε τηλεοράσεις τότε μεσω του κατάλληλου κουμπιού μεταβαινουμε στη κλάση **printTVS** . Αρχικά ανοίγουμε το αρχείο tvs.txt , κατόπιν δημιουργούμε ξανά την `arrayList` με τις τηλεοράσεις. Επιπρόσθετα δημιουργούμε ένα `JFrame`, ένα `JTextArea` και ένα `JScrollPane`. Τέλος χρησιμοποιώντας την `arrayList` και ένα αντικείμενο τύπου `TVClass` μπορούμε με κατάλληλη μορφοποίηση να παρουσιάσουμε τις τηλεοράσεις στην οθόνη(**Figure: 53**)



Figure 53: Εκτύπωση τηλεοράσεων

Ενώ ο κώδικας φαίνεται παρακάτω(**Figure: 54**)

```

-----,
String tvMonitorSize = stringArr[3];
String tvPrice = stringArr[4];
String tvStoreName = stringArr[5];
TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);
tvs_list.add(tv);
}
}
JFrame frame = new JFrame("Print TVS");
frame.setLayout(new BorderLayout());
frame.setSize(800, 500);
JTextArea textArea = new JTextArea();
frame.add(textArea);
JScrollPane sp = new JScrollPane(textArea);
textArea.setEditable(false);
JScrollPane scroll = new JScrollPane(textArea,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
frame.add(scroll);
textArea.append("-----TELEVISIONS-----");
textArea.append("\n");
for (TVClass tv : tvs_list) {
    textArea.append("*****" + "\n");
    textArea.append("TV Model : " + tv.getTv_model() + "\n");
    textArea.append("TV Type : " + tv.getTv_type() + "\n");
    textArea.append("TV Year of Release: " + tv.getTv_year() + "\n");
    textArea.append("TV Monitor Size : " + tv.getTv_monitor_size() + "\n");
    textArea.append("TV Price : " + tv.getTv_price() + "\n");
    textArea.append("PC Store Name : " + tv.getStore_name() + "\n");
    textArea.append("\n");
}
}

```

Figure 54: Κώδικας εκτύπωσης τηλεοράσεων

### 4.3.2 SHOPPING BASKET

Όσο αφορά το καλάθι αγορών αρχικά μεταβαίνουμε στη κλάση `ShoppingBasket` η οποία παίρνει ως όρισμα το `username` του πελάτη που έχει εισέλθει στο σύστημα. Η κλάση αυτή υλοποιεί ένα `JFrame` που περιέχει 3 κουμπιά. Ένα που υλοποιεί την λειτουργία προσθήκη στο καλάθι, ένα άλλο που εκτυπώνει το καλάθι αγορών και ένα τελευταίο που απλά είναι το `dispose` του `frame` αυτού. Το `JFrame` φαίνεται παρακάτω (**Figure: 55**)

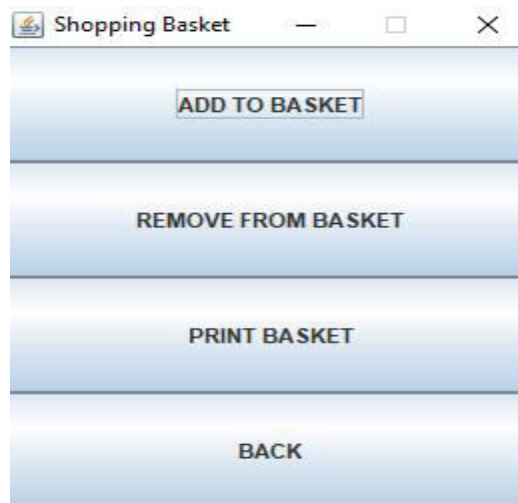


Figure 55: JFrame για το καλάθι αγορών

Ενώ το `frame` της προσθήκης στο καλάθι αγορών (**Figure: 56**)

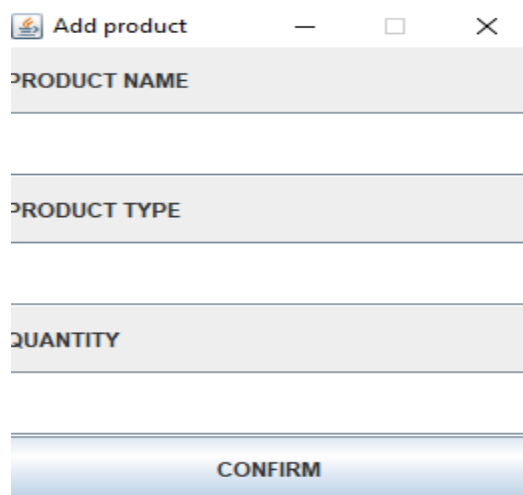


Figure 56: Προσθήκη προϊόντος στο καλάθι αγορών

Εδώ, έχοντας λαβεί το μοντέλο του προϊόντος, το είδος και την ποσότητα κάθε φορά με ένα κατάλληλο `if` μεταβαίνουμε στις συναρτήσεις:

- `shoppingBasketProcessingCamerasProducts`
- `shoppingBasketProcessingConsolesProducts`
- `shoppingBasketProcessingMobilesProducts`
- `shoppingBasketProcessingPCSPProducts`
- `shoppingBasketProcessingTVSPProducts`

Για παράδειγμα η συνάρτηση `shoppingBasketProcessingPCSPProducts` ανοίγει το αρχείο `personalComputers.txt`, δημιουργεί ξανά τη `arrayList` από το αρχείο και κάνοντας τον κατάλληλο έλεγχο για το εάν το προϊόν υπάρχει στο κατάστημα ή και όχι. Τελικά μέσω αυτών των συναρτήσεων καλούνται οι παραάτω συναρτήσεις για τελική αποθήκευση του προϊόντος που δημιουργήθηκε:

- `writeCamerasToFileUsingCustomerUsername`
- `writeConsolesToFileUsingCustomerUsername`
- `writeMobilesToFileUsingCustomerUsername`
- `writePCSToFileUsingCustomerUsername`
- `writeTVSToFileUsingCustomerUsername`

### **4.3.3 LOGOUT**

Έξοδος από το σύστημα κάνοντας `dispose` το κεντρικό `JFrame` του `customer panel`.

## 5. ΜΕΛΛΟΝΤΙΚΟ ΕΡΓΟ( FUTURE WORK)

### 5.1 Η ΧΡΗΣΗ ΤΗΣ SQL ΔΙΝΕΙ ΕΥΚΟΛΟΤΕΡΗ ΥΛΟΠΟΙΗΣΗ

Η εφαρμογή μας **στηρίχτηκε καθαρά στην χρήση .txt αρχείων** για την αποθήκευση των δεδομένων των χρηστών αλλά και των προϊόντων του υποκαταστήματος. Θα μπορούσαμε να είχαμε χρησιμοποιήσει βάσεις δεδομένων όπως είναι για παράδειγμα η MySQL ή η PostgreSQL. Χρησιμοποιώντας μια σχεσιακή βάση δεδομένων τα πράγματα είναι πολύ πιο απλά και ως προς το κομμάτι του κώδικα αλλά και προς την υλοποίηση του από τον προγραμματιστή(MySQL)

Για παράδειγμα θα μπορούσαμε αντί για ένα .txt χρηστών( παρόμοιος για όλα τα .txt της εφαρμογής μας) να είχαμε δημιουργήσει μια βάση δεδομένων είτε χειροκίνητα κατεβάζοντας την εφαρμογή MySQL Community είτε μέσω της MySQL Workbench πολύ απλά γράφοντας την παρακάτω εντολή στη μορφή MySQL :

```
CREATE DATABASE users;
```

Όπου το users είναι το όνομα του πίνακα που θέλουμε να δημιουργήσουμε ενώ το CREATE DATABASE είναι δεσμευμένες εντολές της SQL για δημιουργία μιας βάσης. Για να την χρησιμοποιήσουμε πολύ απλά γράφουμε την εντολή(MySQL):

```
USE users;
```

Μεσα στη βάση μπορούμε για παράδειγμα να δημιουργήσουμε διάφορα tables. Για παράδειγμα ένας user έχει ονοματεπώνυμο, αριθμό τηλεφώνου, password και username. Αυτό για παράδειγμα θα μπορούσε να γίνει ως εξής: (MySQL)

```
CREATE TABLE user
```

```
(
```

```
id INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the record
```

```
fname VARCHAR(150) NOT NULL, # First name of the user
```

```
surname VARCHAR(150) NOT NULL, # Surname of the user
```

```
telephone VARCHAR(10),NOT NULL # Birthday of the cat
```

```
password VARCHAR(30),NOT NULL # Password of the user
```

```
PRIMARY KEY (id) # Make the id the primary key
```



);

Το INT χρησιμοποιείται για να δηλώσουμε ακεραίους, το varchar συμβολοσειρές ενώ σε κάθε table χρειαζόμαστε ένα μοναδικό κλειδί όπου εδώ είναι το id του χρήστη. Με παρόμοιο τρόπο θα μπορούσαμε να φτιάξουμε βάσεις και κατάλληλα tables σε κάθε βάση για κάθε .txt αρχείο μας όπου με τη χρήση της SQL αλλά και της Java να δημιουργούσαμε μια πιο σύγχρονη και εύκολη προς υλοποίηση εφαρμογή(MySQL).

## 5.2 Η ΣΥΝΔΕΣΗ ΤΗΣ SQL ΜΕ ΤΗΝ JAVA

Αρχικά πρέπει να κατεβάσουμε τον κατάλληλο driver για να γίνει η σύνδεση μας πχ θα μπορούσαμε να κατεβάσουμε τον JDBC Driver της SQL από το παρακάτω link:

<https://www.microsoft.com/en-us/download/details.aspx?id=11774>

Στη συνέχεια να **τον κάνουμε add** στις βιβλιοθήκες μας πχ στο Netbeans. Κατόπιν με τις παρακάτω εντολές έχουμε σύνδεση στη βάση μας :

```
public static void main (String[] args) throws Exception{  
  
    Connection conn=null;  
  
    String url="jdbc:jtds:sqlserver://YourServerIp:1433/dbName";  
  
    String username="some_username";  
  
    String password="some_password";  
  
    String driver="net.sourceforge.jtds.jdbc.Driver";  
  
    Class.forName(driver);  
  
    conn= DriverManager.getConnection(url, username, password);  
  
}
```

Όπου dbName είναι το όνομα της βάσης μας ενώ username, password είναι τα κατάλληλα username και password που χρειάζονται για την είσοδο μας στη βάση μας. Κάνουμε την σύνδεση κάθε φορά που τη χρειαζόμαστε (πχ σε ανάλογη κλάση) και στη συνέχεια γράφουμε τα κατάλληλα υπερωτήματα( queries) για να πάρουμε ή να στείλουμε μια πληροφορία στη βάση μας, σε συγκεκριμένο table , όπου αυτό χρειάζεται.

### 5.3 SQL – JAVA ΥΠΕΡΩΤΗΜΑΤΑ(QUERIES)

Για να κάνουμε ένα κατάλληλο query πρέπει να κάνουμε τα παρακάτω βήματα(Alvin, 2018):

- Αρχικά δημιουργούμε ένα JDBC ResultSet αντικείμενο
- Εκτελούμε το SQL query SELECT
- Λαμβάνουμε τα κατάλληλα αποτελέσματα

Εάν για παράδειγμα θέλουμε να βρούμε τα προϊόντα που είναι τύπου υπολογιστή και έχουν τύπο μοντέλου Alienware και βρίσκονται στην εταιρεία Κωτσόβολος κάνουμε τα εξής παρακάτω(Alvin, 2018):

```
Statement stmt = conn.createStatement();
```

```
ResultSet rs = stmt.executeQuery("SELECT Alienware FROM personalComputers WHERE company=KOTSOVLOS" )
```

Με παρόμοιο τρόπο μπορούμε θέσουμε ερωτήματα στη βάση εφόσον φυσικά είναι συνδεδεμένη με την εφαρμογή μας και να παίρνουμε τις κατάλληλες απαντήσεις. Μέσω αυτών των κατάλληλων απαντήσεων και πράξεων μπορούμε να στείλουμε τις αλλαγμένες τιμές ή νέες τιμές πίσω στη βάση.

Η διαδικασία αποστολής δεδομένων πίσω στη βάση χρειάζεται και 2 βήματα(Alvin, 2018):

- Δημιουργούμε ένα Java statement αντικείμενο
- Εκτελούμε ένα SQL query τύπου INSERT διαμέσου του JDBC αντικειμένου

Οποτε για το παραπάνω παράδειγμα η εισαγωγή ενός Alienware στο κατάλληλο table της βάσης μας γίνεται ως εξής(Alvin, 2018):

```
Statement = conn.createStatement();
```

```
statement.executeUpdate("INSERT INTO personalComputers " + "VALUES (Alienware, i7, 3.2, GTX180Ti, 2000, 64, 2850, KOTSOVOLOS)");
```

## **ΣΥΜΠΕΡΑΣΜΑΤΑ**

Στη σημερινή εποχή είναι απαραίτητο να μπορούμε όσο το δυνατόν να γλιτώνουμε περιττό χρόνο μιας καθώς ζούμε σε πολύ γρήγορους ρυθμούς ζωής. Οπότε η χρησιμότητα εφαρμογών που σκοπεύουν στην αγοραπωλησία προϊόντων και υπηρεσιών εξ αποστάσεως είναι μια καλή ιδέα ώστε ο μέσος άνθρωπος να μπορέσει έστω να γλιτώσει την ταλαιπωρία επίσκεψης του σε κάποιο κατάστημα. Επομένως, η αγορά από το διαδίκτυο είναι γρήγορη, εύκολη και αρκετά αξιόπιστη.

Με το πάτημα ενός κουμπιού μπορεί η παραγγελία σου να έρθει στο σπίτι, στο γραφείο σου χωρίς περιττές βόλτες και το βαναυσο χάσιμο χρόνου. Ένα όμως θέμα σοβαρό που τίγεται όσο αφορά τα ηλεκτρονικά καταστήματα είναι η ασφάλεια τους. Δίνοντας τα προσωπικά σου στοιχεία και πιθανόν τον αριθμό της χρεωστικής/πιστωτικής σου κάρτας, δεν μπορείς ποτέ να είσαι 100% σίγουρος ότι θα είσαι ασφαλής. Οπότε, κύριο μέλημα των επιστημόνων πληροφορικής που ασχολούνται με τέτοια θέματα είναι να μπορέσουν να αυξήσουν την ασφάλεια στα ηλεκτρονικά καταστήματα.

## ΑΝΑΦΟΡΕΣ

- ✓ Chisholm, H. (1911). Commerce- Encyclopædia Britannica. 6 (11th). Cambridge University Press
- ✓ Khurana, A.(2017).Advantages of E-commerce Over Traditional Retail. Retrieved from [here](#) [accessed 22 April 2018]
- ✓ Khurana, A.(2018).Disadvantages of E-commerce. Retrieved from [here](#) [accessed 22 April 2018]
- ✓ Manthena, P.(2013). Connecting a Java application to an SQL database with Eclipse, Stackoverflow. Retrieved from [here](#) [accessed 18 June 2018]
- ✓ MySQL, Creating and Selecting a Database. Retrieved from [here](#) [ accessed 18 June 2018]
- ✓ MySQL, Some Basic Operations with MySQL. Retrieved from [here](#) [ accessed 18 June 2018]
- ✓ Wienclaw, Ruth A. (2013).E-Commerce. Research Starters.
- ✓ Wills, B.(2014).18 Major Benefits of e-Commerce Business for Retailers & Customers 2018. Retrieved from [here](#) [accessed 22 April 2018]
- ✓ Alvin, A.(2018). Java JDBC: A SQL SELECT query example. Retrieved from [here](#) [ accessed 22 June 2018]
- ✓ Alvin, A.(2018). Java JDBC Insert Example: How to insert data into a SQL table. Retrieved from [here](#) [accessed 22 June 2018]