



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΕΚΔΟΣΗΣ ΠΑΡΑΣΤΑΤΙΚΩΝ ΓΙΑ**  
**ΕΜΠΟΡΙΚΗ ΕΠΙΧΕΙΡΗΣΗ**

**Μάριος Χ. Πάτσης**

**Εισηγητής: Νικόλαος Ζάχαρης**

**ΑΘΗΝΑ 2018**



Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση  
Μάριος Χ. Πάτσης  
Α.Μ. 41790**

**Εισηγητής:**

**Δρ. Νικόλαος Ζάχαρης, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης ...../..... /2018**



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Πάτσης Μάριος του Χρήστου, με αριθμό μητρώου 41790 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών. του Πανεπιστημίου Δυτικής Αττικής πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε με τη βοήθεια του επιβλέπων καθηγητή μου, τον οποίο θα ήθελα να ευχαριστήσω πολύ.

Θερμές ευχαριστίες θα ήθελα να δώσω σε όλους τους καθηγητες που είχα όλα τα χρόνια της μέχρι στιγμής ακαδημαϊκής μου ζωής, για τις γνώσεις που μου έδωσαν.





## Περίληψη

Η αντικατάσταση του χειρόγραφου που χρησιμοποιούσαν επί δεκαετίες με τη μηχανογράφηση που χρησιμοποιείται σήμερα, έλυσε τα προβλήματα που υπήρχαν από τη χρήση του χειρόγραφου.

Η έρευνα έχει καταδείξει πως οι περισσότερες από τις μεγάλες επιχειρήσεις στη δομή και τη λειτουργία των τμημάτων τους χρησιμοποιούν εμπορικά προγράμματα και ωφελούνται από αυτό καθώς έχουν μεγαλύτερη απόδοση λόγω της ενημέρωσης για την κατάσταση της επιχείρησής τους και αυτό βοηθάει την ορθή λήψη αποφάσεων, της εξοικονόμησης του χώρου αλλά και του χρόνου λόγω της ταχύτητας αλλά και της άμεσης επικοινωνίας των επιμέρους λειτουργιών.

Σκοπός της εργασίας είναι η δημιουργία μιας εφαρμογής έκδοσης τιμολογίων ώστε να δείξουμε τα πλεονεκτήματα χρήσης μιας τέτοιας εφαρμογής από επιχειρήσεις.



## **Abstract**

Replacing the handwriting used for decades with the computerization used today solved the problems that arose from using the handwriting.

Research has shown that most of the large companies in the structure and operation of their departments use commercial programs and benefit from it as they have a higher return on account of their business situation and this helps in making decisions, time due to the speed and direct communication of the individual functions.

The purpose of this study is the development of an application issuing invoices, in order to show the great advantages of using a computerized application, instead of handwriting.

Science Field: Software development

Keywords: Electronic invoicing, Java, MySQL, invoice,application



## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>19</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....</b>	<b>21</b>
2.1 Οι Εμπορικές Επιχειρήσεις .....	21
2.2 Τα Παραστατικά των Επιχειρήσεων .....	21
2.3 Η αντικατάσταση των χειρόγραφων τιμολογίων.....	22
2.3 Ηλεκτρονική τιμολόγηση .....	23
2.4 Η Διανομή .....	25
2.5 Χρήση ηλεκτρονικής τιμολόγησης στη χώρα μας .....	26
2.6 Η Γλώσσα Java.....	27
2.7 MySQL .....	29
<b>ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>33</b>
3.1 Παρόμοιες εφαρμογές.....	33
3.2 Εργαλεία ανάπτυξης εφαρμογής.....	35
3.2.1 Το Eclipse IDE.....	35
3.2.2 Δομή κώδικα εφαρμογής.....	35
3.2.3 Τα δεδομένα της εφαρμογής.....	37
3.2.4 Δημιουργία εγγράφων PDF.....	39
3.2.5 Υπηρεσία ΓΓΠΣ για αναζήτηση – έλεγχο ΑΦΜ.....	39
<b>ΕΡΕΥΝΗΤΙΚΟ ΚΟΜΜΑΤΙ .....</b>	<b>41</b>
4.1 Αρχική οθόνη.....	41
4.2 Ρυθμίσεις.....	41
4.2.1 Στοιχεία επιχείρησης.....	42
4.2.2 Email.....	44

<b>4.3</b>	<b>Επαφές</b> .....	<b>46</b>
4.3.1	Εμφάνιση επαφής .....	47
4.3.2	Τροποποίηση στοιχείων .....	49
4.3.3	Νέα επαφή .....	51
<b>4.4</b>	<b>Προϊόντα</b> .....	<b>55</b>
4.4.1	Εμφάνιση προϊόντος .....	57
4.4.2	Τροποποίηση προϊόντος .....	60
4.4.3	Δημιουργία προϊόντος .....	61
<b>4.5</b>	<b>Τιμολόγια</b> .....	<b>63</b>
4.5.1	Πληρωμή.....	63
4.5.3	Νέο τιμολόγιο .....	70
<b>4.6</b>	<b>Αναφορές</b> .....	<b>74</b>
4.6.1	Αναφορές επαφής .....	76
4.6.2	Αναφορές πωλήσεων.....	78
4.6.3	Αναφορές προϊόντων .....	79
<b>ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ</b> .....		<b>83</b>
<b>5.1</b>	<b>Δημιουργία βάσης δεδομένων</b> .....	<b>83</b>
<b>5.2</b>	<b>Πακέτο Util</b> .....	<b>86</b>
5.2.1	Main.java .....	86
5.2.2	MySqlConnection .....	89
<b>5.3</b>	<b>Πακέτο contacts</b> .....	<b>90</b>
5.3.1	Erafes.java .....	90
5.3.2	NewContact.java .....	94
5.3.3	ContactPG.java.....	102
5.3.4	ContactsD.java .....	108
<b>5.4</b>	<b>Πακέτο products</b> .....	<b>118</b>
5.4.1	ProductsG.java .....	118
5.4.2	NewProductG.java .....	122
5.4.3	ProductPG.java .....	126
5.4.4	ProductsD.java .....	133
<b>5.5</b>	<b>Πακέτο products</b> .....	<b>142</b>
5.5.1	InvoiceG.java.....	142

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

5.5.2 NewInvoice.java .....	148
5.5.3 InvoicePG.java .....	163
5.5.4 InvoiceD.java .....	171
<b>5.6 Πακέτο reports .....</b>	<b>186</b>
5.6.1 ReportsG.java .....	186
<b>5.7 Πακέτο settings .....</b>	<b>203</b>
5.7.1 SettingsG.java .....	203
5.7.2 Email.java .....	215
5.7.3 GenerateInvoice.java .....	218
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>227</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>229</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1 ΑΠΟΣΤΟΛΗ ΤΙΜΟΛΟΓΙΟΥ ΜΕ EMAIL .....	25
ΕΙΚΟΝΑ 2.1 ΤΟ ΛΟΓΟΤΥΠΟ ΤΗΣ ΓΛΩΣΣΑΣ JAVA. ....	27
ΕΙΚΟΝΑ 2.2 ΤΟ ΛΟΓΟΤΥΠΟ ΤΗΣ MYSQL. ....	29
ΕΙΚΟΝΑ 3.1 ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ ΕΦΑΡΜΟΓΗΣ ELORUS .....	33
ΕΙΚΟΝΑ 3.2 ΟΘΟΝΗ ΠΕΛΑΤΟΛΟΓΙΟΥ .....	34
ΕΙΚΟΝΑ 3.3 ΟΘΟΝΗ ΠΡΟΪΟΝΤΑ.....	34
ΕΙΚΟΝΑ 2.4 ΤΟ ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΤΟΥ ECLIPSE IDE.....	35
ΕΙΚΟΝΑ 3.5 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ.....	36
ΕΙΚΟΝΑ 3.6 ΚΛΑΣΕΙΣ ΓΙΑ ΑΡΧΙΚΗ ΟΘΟΝΗ ΚΑΙ ΣΥΝΔΕΣΗ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ .....	36
ΕΙΚΟΝΑ 3.7 ΚΛΑΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑΦΩΝ.....	36
ΕΙΚΟΝΑ 3.8 ΚΛΑΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΪΟΝΤΩΝ .....	36
ΕΙΚΟΝΑ 3.9 ΚΛΑΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΙΜΟΛΟΓΙΩΝ .....	36
ΕΙΚΟΝΑ 3.10 ΚΛΑΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΡΥΘΜΙΣΕΩΝ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ PDF ΤΙΜΟΛΟΓΙΟΥ.....	37
ΕΙΚΟΝΑ 3.11 ΚΛΑΣΕΙΣ ΓΙΑ ΑΝΑΦΟΡΕΣ .....	37
ΕΙΚΟΝΑ 3.12 ΚΛΑΣΕΙΣ ΣΥΝΔΕΣΗΣ ΕΦΑΡΜΟΓΗΣ ΜΕ ΓΓΠΣ .....	37
ΕΙΚΟΝΑ 3.13 ΣΧΗΜΑΤΙΚΟ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΦΑΡΜΟΓΗΣ .....	38
ΕΙΚΟΝΑ 4.1 ΑΡΧΙΚΗ ΟΘΟΝΗ.....	41
ΕΙΚΟΝΑ 4.2 ΟΘΟΝΗ ΑΛΛΑΓΗΣ ΣΤΟΙΧΕΙΩΝ ΕΠΙΧΕΙΡΗΣΗΣ .....	42
ΕΙΚΟΝΑ 4.3 ΣΥΜΠΛΗΡΩΜΕΝΗ ΟΘΟΝΗ ΕΙΣΑΓΩΓΗΣ ΣΤΟΙΧΕΙΩΝ ΕΠΙΧΕΙΡΗΣΗΣ.....	42
ΕΙΚΟΝΑ 4.4 ΠΑΡΑΘΥΡΟ ΕΠΙΛΟΓΗΣ ΕΙΚΟΝΑΣ.....	43
ΕΙΚΟΝΑ 4.5 ΣΥΜΠΛΗΡΩΜΕΝΗ ΟΘΟΝΗ ΕΙΣΑΓΩΓΗΣ ΣΤΟΙΧΕΙΩΝ ΕΠΙΧΕΙΡΗΣΗΣ. ....	44
ΕΙΚΟΝΑ 4.6 ΟΘΟΝΗ ΑΛΛΑΓΗΣ ΡΥΘΜΙΣΕΩΝ EMAIL. ....	45
ΕΙΚΟΝΑ 4.7 ΟΘΟΝΗ ΕΠΑΦΗΣ .....	46
ΕΙΚΟΝΑ 4.8 ΑΝΑΖΗΤΗΣΗ ΕΠΑΦΗΣ .....	47
ΕΙΚΟΝΑ 4.9 ΕΠΙΛΟΓΗ ΕΠΑΦΗΣ ΓΙΑ ΕΜΦΑΝΙΣΗ .....	48
ΕΙΚΟΝΑ 4.10 ΚΑΡΤΕΛΑ ΕΠΑΦΗΣ .....	49
ΕΙΚΟΝΑ 4.11 ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΕΠΑΦΗΣ .....	50
ΕΙΚΟΝΑ 4.12 ΑΛΛΑΓΗ ΗΜΕΡΟΜΗΝΙΑΣ ΤΡΟΠΟΠΟΙΗΣΗΣ ΜΕΤΑ ΑΠΟ ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΕΠΑΦΗΣ. ....	50
ΕΙΚΟΝΑ 4.13 ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ.....	51
ΕΙΚΟΝΑ 4.14 ΣΥΜΠΛΗΡΩΣΗ Α.Φ.Μ ΓΙΑ ΑΝΑΖΗΤΗΣΗ .....	52



ΕΙΚΟΝΑ 4.15 ΕΠΙΤΥΧΗΣ ΑΝΑΖΗΤΗΣΗ ΑΦΜ.....	53
ΕΙΚΟΝΑ 4.16 ΑΥΤΟΜΑΤΗ ΣΥΜΠΛΗΡΩΣΗ ΚΕΛΙΩΝ .....	53
ΕΙΚΟΝΑ 4.17 ΜΗΝΥΜΑ ΠΡΟΣ ΧΡΗΣΤΗ .....	54
ΕΙΚΟΝΑ 4.18 ΝΕΑ ΕΠΑΦΗ ΜΕ ΣΥΜΠΛΗΡΩΜΕΝΑ ΟΛΑ ΤΑ ΑΠΑΙΤΟΥΜΕΝΑ ΚΕΛΙΑ .....	55
ΕΙΚΟΝΑ 4.19 ΟΘΟΝΗ ΠΡΟΪΟΝΤΑ .....	55
ΕΙΚΟΝΑ 4.20 ΑΝΑΖΗΤΗΣΗ ΠΡΟΪΟΝΤΟΣ.....	56
ΕΙΚΟΝΑ 4.21 ΕΜΦΑΝΙΣΗ ΠΡΟΪΟΝΤΟΣ .....	57
ΕΙΚΟΝΑ 4.22 ΚΑΡΤΕΛΑ ΠΡΟΪΟΝ.....	58
ΕΙΚΟΝΑ 4.23 ΑΛΛΑΓΗ ΚΑΤΑΣΤΑΣΗΣ ΑΠΟΘΗΚΗΣ.....	59
ΕΙΚΟΝΑ 4.24 ΠΡΟΪΟΝ ΕΝΕΡΓΟ - ΑΝΕΝΕΡΓΟ .....	60
ΕΙΚΟΝΑ 4.25 ΤΡΟΠΟΠΟΙΗΣΗ ΠΡΟΪΟΝΤΟΣ .....	61
ΕΙΚΟΝΑ 4.26 ΟΘΟΝΗ ΝΕΟ ΠΡΟΪΟΝ .....	62
ΕΙΚΟΝΑ 4.27 ΣΥΜΠΛΗΡΩΣΗ ΚΕΛΙΩΝ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΚΑΙΝΟΥΡΓΙΟΥ ΠΡΟΪΟΝΤΟΣ.....	62
ΕΙΚΟΝΑ 4.28 ΟΘΟΝΗ ΤΙΜΟΛΟΓΙΑ.....	63
ΕΙΚΟΝΑ 4.29 ΠΛΗΡΩΜΗ ΤΙΜΟΛΟΓΙΟΥ .....	64
ΕΙΚΟΝΑ 4.30 ΑΛΛΑΓΗ ΣΕ ΜΗ ΠΛΗΡΩΜΕΝΟ.....	65
ΕΙΚΟΝΑ 4.31 ΟΘΟΝΗ ΤΙΜΟΛΟΓΙΟ .....	66
ΕΙΚΟΝΑ 4.32 ΑΠΟΘΗΚΕΥΣΗ ΣΕ ΑΡΧΕΙΟ .....	67
ΕΙΚΟΝΑ 4.33 ΑΠΟΣΤΟΛΗ ΣΕ EMAIL .....	67
ΕΙΚΟΝΑ 4.34 ΔΕΝ ΥΠΑΡΧΕΙ EMAIL.....	68
ΕΙΚΟΝΑ 4.36 ΤΟ ΤΙΜΟΛΟΓΙΟ .....	69
ΕΙΚΟΝΑ 4.35 EMAIL ΜΕ ΣΥΝΗΜΜΕΝΟ.....	69
ΕΙΚΟΝΑ 4.37 ΟΘΟΝΗ ΝΕΟ ΤΙΜΟΛΟΓΙΟ.....	70
ΕΙΚΟΝΑ 4.38 ΑΝΕΝΕΡΓΟ ΠΡΟΪΟΝ .....	71
ΕΙΚΟΝΑ 4.40 ΚΑΤΑΧΩΡΗΣΗ ΕΠΙΛΕΓΜΕΝΟΥ ΠΡΟΪΟΝΤΟΣ.....	72
ΕΙΚΟΝΑ 4.39 ΕΠΙΛΟΓΗ ΠΡΟΪΟΝΤΟΣ.....	72
ΕΙΚΟΝΑ 4.41 ΠΡΟΪΟΝ ΜΕ ΑΠΟΘΕΜΑ .....	73
ΕΙΚΟΝΑ 4.42 ΚΑΤΑΧΩΡΗΣΗ ΔΕΥΤΕΡΟΥ ΠΡΟΪΟΝΤΟΣ.....	74
ΕΙΚΟΝΑ 4.43 ΑΠΟΘΗΚΕΥΣΗ ΤΙΜΟΛΟΓΙΟΥ.....	74
ΕΙΚΟΝΑ 4.44 ΟΘΟΝΗ ΑΝΑΦΟΡΕΣ .....	75
ΕΙΚΟΝΑ 4.45 ΑΝΑΦΟΡΕΣ ΕΠΑΦΩΝ.....	76
ΕΙΚΟΝΑ 4.46 ΥΠΟΛΟΙΠΟ ΕΠΑΦΩΝ .....	77
ΕΙΚΟΝΑ 4.47 ΜΕΓΑΛΥΤΕΡΟ ΠΟΣΟ .....	77
ΕΙΚΟΝΑ 4.48 ΑΝΑΦΟΡΕΣ ΠΩΛΗΣΕΩΝ .....	78

ΕΙΚΟΝΑ 4.49 ΠΑΡΑΣΤΑΤΙΚΑ ΠΩΛΗΣΕΩΝ.....	79
ΕΙΚΟΝΑ 4.50 ΑΝΑΦΟΡΕΣ ΠΡΟΪΟΝΤΩΝ .....	80
ΕΙΚΟΝΑ 4.51 ΠΩΛΗΣΕΙΣ ΠΡΟΪΟΝΤΩΝ.....	81

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΕΕ	ΕΥΡΩΠΑΙΚΗ ΕΝΩΣΗ
ΦΠΑ	ΦΟΡΟΣ ΠΡΟΣΤΙΘΕΜΕΝΗΣ ΑΞΙΑΣ
ΔΑ	ΔΕΛΤΙΟ ΑΠΟΣΤΟΛΗΣ
B2B	BUSINESS TO BUSINESS
ERP	ENTERPRISE RESOURCE PLANNING
PDF	PORTABLE DOCUMENT FORMAT
HTTP	HYPERTEXT TRANSFER PROTOCOL
B2C	BUSINESS-TO-CONSUMER
RDBMS	RELATIONAL DATABASE MANagements SYSTEMS
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
EMAIL	ELECTRONIC MAIL
E-INVOICE	ELECTRONIC INVOICE
Η/Υ	ΗΛΕΚΤΡΟΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΗΣ



## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Οι απαιτήσεις της σύγχρονης εποχής έχουν θέσει τη δεξιότητα στη χρήση της πληροφόρησης ως τον σπουδαιότερο παράγοντα για την επιτυχία όλων των επαγγελματιών. Όλοι οι επαγγελματίες επιβάλλεται να γνωρίζουν τον τρόπο απόκτησης πληροφοριών, τη χρήση του, τη διαχείριση, τη διάδοση και την παρουσίαση τους σε άλλα άτομα [12]. Η βάση για όλες τις δραστηριότητες που σχετίζονται με την πληροφόρηση είναι οι ηλεκτρονικοί υπολογιστές και το σύνολο των συνοδευτικών συστημάτων. Η λειτουργία εκατομμυρίων συστημάτων Η/Υ παγκοσμίως εξασφαλίζει την εκτέλεση εκατοντάδων εργασιών. Η ύπαρξη επαγγελματικών κλάδων που δεν είναι απαραίτητη η χρήση Η/Υ είναι ελάχιστη και μελλοντικά θα λιγοστεύει περισσότερο. Η χρήση τους εκ μέρους των επιχειρήσεων, δημοσίων υπηρεσιών και άλλων οργανισμών είναι ευρύτατη γιατί μέσω της τεχνολογίας έχουν λιγότερες ώρες γραφικής εργασίας, εξυπηρετούν καλύτερα τους πελάτες και οι προϊστάμενοι δέχονται βοήθεια για τη λήψη καλύτερων αποφάσεων. Η αντικατάσταση του χειρόγραφου που χρησιμοποιούσαν επί δεκαετίες με τη μηχανογράφηση που χρησιμοποιείται σήμερα, έλυσε τα προβλήματα που υπήρχαν από τη χρήση του χειρόγραφου. Ορισμένα από αυτά τα προβλήματα είναι:

1. Οι υπάλληλοι είναι διαρκώς απασχολημένοι με τις ανάγκες του λογιστηρίου και δεν υπάρχει ο χρόνος για να παρέχουν στα άτομα της διοίκησης πρόσθετες πληροφορίες για να αναλυθούν.
2. Η έκδοση των εκθέσεων παρουσιάζει ανακριβή δεδομένα αφού δεν υπάρχει σωστή εικόνα για την επιχείρηση.
3. Λόγω φόρτου εργασίας οι εκθέσεις αργούν να υποβληθούν και δεν είναι δυνατή η έγκαιρη αξιοποίηση τους εκ μέρους της διοίκησης.
4. Εξαιτίας του ότι με το χειρόγραφο σύστημα χρειάζεται πολύς χρόνος για τη διεκπεραίωση υπάρχει καθυστέρηση και οι πελάτες δεν εξυπηρετούνται σωστά.
5. Τα αποθέματα δεν είναι δυνατόν να ελεγχθούν με ταχύτητα και λεπτομερώς με συνέπεια την ύπαρξη αυξημένου κόστους, σοβαρών ελλείψεων και εξάντλησης σε βασικά προϊόντα.
6. Η έκδοση των λογαριασμών γίνεται με καθυστέρηση και αυξάνεται η πιθανότητα να υπάρχουν επισφαλείς λογαριασμοί.
7. Για τις ενέργειες καταγραφής των στοιχείων επεξεργασίας και απορροής στα αποτελέσματα απαιτείται μεγάλος αριθμός από υπαλλήλους με αποτέλεσμα να αυξάνεται το κόστος του προσωπικού και οι δαπάνες.

8. Τα πολλά έγγραφα είναι αιτία σύγχυσης και άγχους των υπαλλήλων ενώ χρειάζεται η ύπαρξη μεγάλου χώρου για την αρχειοθέτηση, με συνέπεια να αυξάνεται το κόστος και οι δαπάνες.

Μέσω της εργασίας αυτής θα γίνει προσπάθεια μείωσης των παραπάνω προβλημάτων παρουσιάζοντας πως εκδίδονται και διαχειρίζονται τα τιμολόγια, αυτοματοποιημένα.

## ΚΕΦΑΛΑΙΟ 2

### ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

#### 2.1 Οι Εμπορικές Επιχειρήσεις

Ως επιχειρήσεις χαρακτηρίζονται οι οικονομικές μονάδες που έχουν αυτοτέλεια και οργανώνουν υπεύθυνα παραγωγικούς συντελεστές ενώ επίσης διαχειρίζονται συναλλαγές μέσω των οποίων επιδιώκεται η μέγιστη δυνατή απόκτηση κέρδους [7].

Το σύνολο των κυριότερων χαρακτηριστικών για μια επιχείρηση και τα οποία είναι και οι απαραίτητοι όροι έχει ως ακολούθως :

1. Η προϋπόθεση μόνιμου συνδυασμού σε συντελεστές παραγωγής.
2. Να έχει αυτοτέλεια.
3. Οτιδήποτε παράγει να αφορά ομάδες άγνωστου καταναλωτικού κοινού και
4. Να είναι αποτέλεσμα προηγούμενης φάσης ωστόσο η διαμόρφωση της να αφορά συγκεκριμένο κοινωνικοοικονομικό χώρο που θα χαρακτηρίζεται και θα αποτιμάται σε χρήματα αλλά και θα επιδιώκεται η δημιουργία κέρδους [10].

#### 2.2 Τα Παραστατικά των Επιχειρήσεων

Η έννοια του τιμολογίου αφορά το είδος του στοιχείου του οποίου η έκδοση υπόκειται σε νόμους και είναι υποχρεωτικό στις πωλήσεις προϊόντων καθώς και παροχής σε υπηρεσίες, οι οποίες πραγματοποιούνται μέσα στη χώρα ή σε χώρα της ΕΕ ή αλλού αλλά και κάθε φορά που η συναλλαγή επιβάλλει ΦΠΑ από τη νομοθεσία που ισχύει. Τα έγγραφα στα οποία αναγράφεται η πληροφόρηση που είναι απαραίτητη για να εκδοθεί τιμολόγιο έχει την ισχύ τιμολογίου αν αυτός που θα παραλάβει τα αγαθά το αποδεχθεί. Συνοπτικά αναφέρουμε πως τα τιμολόγια εκδίδονται στις ακόλουθες περιπτώσεις [1] :

- α) όταν πωλούνται αγαθά ιδίων ή τρίτων

β) όταν παρέχονται υπηρεσίες

γ) όταν πωλούνται αγαθά ή παρέχονται υπηρεσίες σε άλλη χώρα από τη δική μας

δ) στις περιπτώσεις εκπτώσεων, επιστροφών ή άλλων διαφορών που προκύπτουν όταν πωλούνται αγαθά ή παρέχονται υπηρεσίες.

ε) όταν αγοράζονται αγαθά από μη υπόχρεους (αυτοτιμολόγηση) [3]

Σημειώνεται πως στην πραγματοποίηση οικονομικής δοσοληψίας γίνεται έκδοση τιμολογίου από όποιον υποχρεούται και η διάκριση τους π.χ Τιμολόγιο Πώλησης, Αγορά, Παροχής Υπηρεσιών κ.λ.π ορίζεται από την ανάγκη της κάθε συναλλαγής. Βάσει των νέων διατάξεων ένα τιμολόγιο δεν χρειάζεται θεώρηση είτε είναι χειρόγραφο ή μηχανογραφικό ή ηλεκτρονικό [3].

Οι επιχειρήσεις λοιπόν που δραστηριοποιούνται στις πωλήσεις προϊόντων ή παρέχουν υπηρεσίες έχουν την υποχρέωση έκδοσης τιμολογίου για όλες τις πωλήσεις.

Ένα άλλο είδος είναι το παραστατικό του πιστωτικού τιμολογίου το οποίο εκδίδεται σε περίπτωση έκπτωσης, επιστροφής ή άλλης διαφοράς όπως ελλείμματος κατά τη διάρκεια της αποστολής και της παράδοσης των προϊόντων, λάθος υπολογισμός στην αξία, λάθος επωνυμία, λάθος χρέωση ΦΠΑ κ.ά με τα οποία μπορεί να επηρεαστεί οτιδήποτε στοιχείο περιέχει το αρχικό τιμολόγιο. Εκδίδεται από τους πωλητές αγαθών η από όσους παρέχουν υπηρεσίες [9] .

### **2.3 Η αντικατάσταση των χειρόγραφων τιμολογίων**

Μια επιχείρηση κατά την δραστηριότητα της πρέπει να φυλάει τα τιμολόγια τα οποία έχει εκδώσει και να μπορεί να τα αναζητήσει ευκολά και γρηγορά. Με το χειρόγραφο τιμολόγιο κάτι τέτοιο απαιτεί πολύ χρόνο, χώρο και εργατικούς πόρους, αφού χωρίς την χρήση ηλεκτρονικού υπολογιστή η αποθήκευση και η σωστή ταξινόμηση αυτών των εγγράφων καθίσταται περίπλοκη. [2]

Για την έκδοση ενός τιμολογίου εκτός από τα στοιχεία αγοραστή που απαιτούνται υπάρχουν και τα προϊόντα ή υπηρεσίες που θα αναγράφονται πάνω και είναι και ο λόγος που υπάρχει το τιμολόγιο. Αυτά τα στοιχεία σε ένα χειρόγραφο σύστημα θα πρέπει να γράφονται χειροκίνητα κάθε φορά που γίνεται μια πώληση κάνοντας το πολύ χρονοβόρο.[2]



Η αντικατάσταση των χειρόγραφων τιμολογίων με μηχανογραφημένα τιμολόγια μέσω κατάλληλου λογισμικού έχει δώσει πολλές λύσεις στα παραπάνω προβλήματα και έχει κάνει τον τρόπο της τιμολόγησης να έχει επιφέρει όχι μόνο σε οργάνωση και χρήμα αλλά και σε καλύτερη και γρηγορότερη εξυπηρέτηση των πελατών. Έτσι καταλληλά λογισμικά δημιουργούν έντυπα μηχανογραφημένα τιμολόγια και διατηρούν αρχεία σε ψηφιακή μορφή χωρίς να απαιτείται χώρος φύλαξης του αρχείου αφού θα είναι αποθηκευμένο σε ηλεκτρονικά μέσα αποθήκευσης και επίσης λόγω της ηλεκτρονικής μορφής γίνεται γρήγορη και εύκολη αναζήτηση των τιμολογίων. Η εφαρμογή επίσης λειτουργεί σαν πελατολόγιο, αποθήκη προϊόντων κ.α. ώστε να μην χρειάζεται κάθε φορά να γίνεται εισαγωγή των ιδίων στοιχείων.[2]

### **2.3 Ηλεκτρονική τιμολόγηση**

Με την ανάπτυξη του ηλεκτρονικού εμπορίου, δηλαδή την online παραγγελία, την πληρωμή μέσω τραπεζής κ.α. το τιμολόγιο έπρεπε να παραδωθεί μόνο ως φυσικό έντυπο.[13] Με την παραδοσιακή διαδικασία τιμολόγησης υπήρχαν κάποια μειονεκτήματα καθώς βασιζόταν στο χαρτί, με την ηλεκτρονική διαδικασία σκοπός είναι να μειωθούν αυτά τα μειονεκτήματα. Η οδηγία του Συμβουλίου της Ευρωπαϊκής Ένωσης 2001/115/EK, της 20ής Δεκεμβρίου 2001, περιλαμβάνει τους κανόνες της αρχειοθέτησης αλλά και της ίδιας ηλεκτρονικής τιμολόγησης. Επίσης, περιγράφει το πλαίσιο της ηλεκτρονικής τιμολόγησης και αυτό έχει στόχο να απλοποιηθούν, να εκσυγχρονιστούν και να εναρμονιστούν οι όροι της τιμολόγησης που σχετίζονται με τον φόρο προστιθέμενης αξίας. [22]

Για να υλοποιηθεί η ηλεκτρονική τιμολόγηση γίνεται χρήση συγκεκριμένων μέσων όπως είναι η παράσταση και πληρωμή ηλεκτρονικού τιμολογίου, η ηλεκτρονική ανταλλαγή δεδομένων που αφορά τη μεταφορά δεδομένων ηλεκτρονικά από υπολογιστή σε υπολογιστή με τη χρήση συμφωνημένης μορφής και τέλος, ένα άλλο μέσο είναι η διαχείριση επιχειρησιακών πόρων (Enterprise Resource Planning - ERP), υπάρχουν συστήματα στα οποία εντάσσονται διάφορα εργαλεία και λογισμικό το οποίο δύναται να διαχειριστεί τα τιμολόγια σαν μέρος της γενικότερης εταιρικής διαδικασίας. Με την ηλεκτρονική τιμολόγηση διατηρείται το άθικτο το τιμολόγιο καθώς αποτελεί ένα σύνολο δεδομένων ενώ παράλληλα

καταργούνται αρκετά από μειονεκτήματα της παραδοσιακής τιμολόγησης [25] .

Χρησιμοποιούνται ποικίλοι τύποι ηλεκτρονικού τιμολογίου όπως είναι το αδόμητο έγγραφο τιμολογίου (Unstructured invoice document), το τιμολόγιο στην συγκεκριμένη περίπτωση μπορεί να δημιουργηθεί είτε χειροκίνητα είτε αυτόματα από ένα σύστημα και στη συνέχεια να μετατραπεί σε ηλεκτρονικό έγγραφο. Επίσης, με την ηλεκτρονική σάρωση ένα παραδοσιακό τιμολόγιο σε χαρτί μετατρέπεται σε ηλεκτρονικό έγγραφο. Υπάρχει και ο τύπος του δομημένου εγγράφου τιμολογίου (Structured invoice document) όπως είναι το Electronic Data Interchange For Administration, Commerce and Transport (EDIFACT) ή το Extensible Markup Language (XML), το οποίο για να δημιουργηθεί πρέπει να συλλεχθούν τα απαιτούμενα στοιχεία σε μια συμφωνημένη μορφή ηλεκτρονικού τιμολογίου που το περιεχόμενό του, η δομή του αλλά και η μορφή του είναι συμφωνημένη [25] .

Πλεονεκτήματα ηλεκτρονικής τιμολόγησης (e-invoicing)

- Μειώνεται ο χρόνος για την έκδοση, αποστολή, παραλαβή και διαχείριση των τιμολογίων
- Μειώνει τις δαπάνες σε αναλώσιμα υλικά λόγω της μη εκτύπωσης των τιμολογίων
- Μειώνονται τα προβλήματα όπως απώλειες και φθοράς τιμολογίου
- Αποφυγή λαθών από την καταχώρηση δεδομένων παραστατικών σε άλλα λογισμικά πχ μέσω της ηλεκτρονικής ανταλλαγής δεδομένων (EDI)
- Δεν υπάρχει σπάταλη χώρα για την αποθήκευση των τιμολογίων [13]
- Μειώνεται η ρύπανση του περιβάλλοντος
- Άμεση ενημέρωση του κράτους για τους φόρους
- Πιο εύκολος οικονομικός έλεγχος από το υπουργείο στα παραστατικά των επιχειρήσεων [14]

## 2.4 Η Διανομή



Εικόνα 1 Αποστολή τιμολογίου με email

Ένα τιμολόγιο χειρόγραφο ,μηχανογραφημένο ή ηλεκτρονικό πρέπει να εκδίδεται πάντα πριν την έναρξη της διακίνησης των εμπορευμάτων .Αν είναι χειρόγραφο ή μηχανογραφημένο πρέπει το έντυπο τιμολόγιο να συνοδεύει το εμπόρευμα μέχρι να το παραλάβει ο αγοραστής. Αν είναι ηλεκτρονικό τιμολόγιο γίνεται αποστολή απευθείας στον αγοραστή πχ. μέσω ηλεκτρονικού ταχυδρομείου πριν την διακίνηση αλλά θα πρέπει να υπάρχει η δυνατότητα αυτού που μεταφέρει το εμπόρευμα να αποδείξει ότι έχει γίνει ηλεκτρονική έκδοση σε περίπτωση που του ζητηθεί πχ να δείξει το ηλεκτρονικό τιμολόγιο μέσω κινητού τηλεφώνου. [24]

Για να διακινηθούν τα ηλεκτρονικά τιμολόγια χρησιμοποιούνται διάφορα ηλεκτρονικά μέσα όπως είναι, η ηλεκτρονική παράσταση (Electronic presentment), όπου το τιμολόγιο παρουσιάζεται σε ένα διαδικτυακό περιβάλλον άμεσα συνδεδεμένο και δεν θα σταλθεί στον παραλήπτη, ο δέκτης θα ενημερωθεί για τη δημιουργία ενός νέου τιμολογίου λαμβάνοντας μήνυμα ηλεκτρονικού ταχυδρομείου. Επίσης, το άμεσα συνδεδεμένο διαδικτυακό περιβάλλον δίνει τη δυνατότητα να διαχειριστούν και να πληρωθούν τα τιμολόγια με πληρωμή απευθείας μέσω τραπεζής. Ακόμα διακίνηση του τιμολογίου γίνεται μέσω του ηλεκτρονικού ταχυδρομείου, καθώς συνάπτεται σαν PDF ή σαν αρχείο κειμένου [25].

## 2.5 Χρήση ηλεκτρονικής τιμολόγησης στη χώρα μας

Στην Ελλάδα η ηλεκτρονική τιμολόγηση έχει αναπτυχθεί και αυτή η άποψη ενισχύεται καθώς περισσότερες από 3.000 επιχειρήσεις έχουν ενταχθεί σε φορείς που παρέχουν ηλεκτρονική τιμολόγηση υιοθετώντας το συγκεκριμένο πλαίσιο. Αν και σε σχέση με την Ευρώπη η Ελλάδα δεν έχει υιοθετήσει ακόμα την ηλεκτρονική τιμολόγηση αφού μόνο το 2,5% αποτελούν τα ηλεκτρονικά τιμολόγια που διακινούνται στην χώρα μας έναντι του 70% κατά μέσο όρο στην υπόλοιπη Ευρώπη.[23] Το γεγονός πως οι επιχειρήσεις στην Ελλάδα που χρησιμοποιούν την ηλεκτρονική τιμολόγηση ολοένα και αυξάνονται είναι αξιοσημείωτο και οι λόγοι που την υιοθετούν είναι πως το κόστος μειώνεται σημαντικά σχετικά με την επανακαταχώρηση των τιμολογίων από τους παραλήπτες και γενικότερα το πως διαχειρίζεται και διακινείται το έντυπο τιμολόγιο [17].. Ένας άλλος σημαντικός παράγοντας που αξίζει να αναφερθεί είναι πως το κόστος αποθήκευσης και τα έξοδα εκτύπωσης και αποστολής μειώνονται καθώς το χαρτί καταργείται. Στις ελληνικές επιχειρήσεις η παραδοσιακή τιμολόγηση εκτιμάται πως είχε συνολικό κόστος 7,20 ευρώ ανά τιμολόγιο, ενώ με την πλήρη εφαρμογή της ηλεκτρονικής τιμολόγησης το μέσο κόστος είναι στα 2,40, ετσι είναι δυνατόν να εξοικονομηθούν ετησίως 4-5 δισεκατομμύρια ευρώ. Επιπρόσθετα, παρέχεται η δυνατότητα να διενεργούνται πρόσθετοι έλεγχοι μακροπρόθεσμα στο σύνολο των τιμολογίων και ο φορολογικός έλεγχος να είναι ταχύτερος και πιο ουσιαστικός. Το υπουργείο Οικονομικών, δεδομένου του ότι η κύρια στρατηγική που αφορά στο ηλεκτρονικό εμπόριο στην Ευρώπη είναι το ηλεκτρονικό τιμολόγιο, έχει στόχο την πλήρη εφαρμογή της υπουργικής απόφασης που σχετίζεται με τη ηλεκτρονική τιμολόγηση, αξιοποιώντας την εμπειρία τριών χιλιάδων επιχειρήσεων που ήδη έχουν χρησιμοποιήσει την ηλεκτρονική τιμολόγηση όπως επίσης και επιχειρήσεων παροχής σχετικών υπηρεσιών [6].

Επιπλέον, είναι υπό εξέταση το αν θα υιοθετηθεί η ηλεκτρονική τιμολόγηση από το ελληνικό δημόσιο ως αγοραστή, από όλες τις ανώνυμες επιχειρήσεις και γενικά από όλες τις ελληνικές επιχειρήσεις αλλά και από τις επιχειρήσεις του Χρηματιστηρίου Αθηνών. [17].

## 2.6 Η Γλώσσα Java



Εικόνα 2.1 Το λογότυπο της γλώσσας Java.

Τα πιο βασικά χαρακτηριστικά της Java είναι τα ακόλουθα: Τα κυριότερα γνωρίσματα της γλώσσας Java είναι τα παρακάτω [8] :

- Δεν είναι σύνθετη: Υπάρχει αναφορά σε ένα σύνολο πηγών που έχουν σχέση με τη Java για την ύπαρξη του συγκεκριμένου χαρακτηριστικού όμως δεν αποτελεί αντικειμενική εκτίμηση. Η Java έχει τη μορφή της απλής γλώσσας με δεδομένο πάντα το μέγεθος του απλού που έχουν οι γλώσσες για τον προγραμματισμό. Αν συγκριθεί με την C++ είναι πιο απλή αφού δεν χρησιμοποιούνται δείκτες (pointers) και η μνήμη της είναι διαχειριζόμενη εκ μέρους της ίδιας της γλώσσας.

- Μεταγλωττίζεται (compiled) και ερμηνεύεται (interpreted): Αντίθετα με τις C/C++ που είναι μεταγλωττιζόμενες η παραγωγή εκτελέσιμου κώδικα της Java χρησιμοποιεί και τις δύο τεχνικές. Η περιγραφή της διαδικασίας αυτής αναφέρεται αναλυτικότερα σε άλλο σημείο της εργασίας [11].

- Υποστηρίζει τον αντικειμενοστρεφή προγραμματισμό (OOD): Η Java είναι υποστηρίκτρια κατ'αποκλειστικότητα του μοντέλου αντικειμενοστρεφούς προγραμματισμού (object-oriented design) και δεν μπορεί να χρησιμοποιηθεί βάσει κάποιου άλλου μοντέλου για παράδειγμα διαδικαστικού [15].

- Έχει φορητότητα επιπέδου μεταγλωττισμένου κώδικα: Αποτελεί το πιο ισχυρό της γνώρισμα . Ένα από τα πιο ισχυρά χαρακτηριστικά της Java που στην πράξη

εννοεί οι προγραμματιστές μπορούν να γράψουν και να μεταγλωττίσουν ένα πρόγραμμα για παράδειγμα σε Windows και μετά να λάβει το μεταγλωττισμένο αρχείο και να το εφαρμόσει σε μηχάνημα Unix χωρίς να το αλλάξει υπό τον όρο πως στο Unix έχει εγκατασταθεί το JRE που αντιστοιχεί [18].

- Πραγματοποιεί αυστηρούς ελέγχους στους τύπους (strongly typed): Ο προγραμματιστής απαιτείται από την Java να χρησιμοποιεί σωστά τους τύπους και δεν απαγορεύει την όποια αυθαίρετη μετατροπή όπως οι C/C++.

- Έχει υψηλό επίπεδο: Υπάρχει η δυνατότητα εκμάθησης της Java και της σύνταξης κώδικα σχετικά εύκολα αφού χρησιμοποιούνται λέξεις στα Αγγλικά και όχι λέξεις της μηχανικής γλώσσας.

- Είναι εξαιρετικά ασφαλής: Τα προγράμματα που εκτελούνται δέχονται τον έλεγχο μηχανισμών ασφαλείας για να αποτρέψουν κακόβουλους κώδικες.

- Υποστήριξη πολυμέσων: Είναι μία εκ των ελαχίστων γλωσσών στην κατηγορία της που υποστηρίζει τις εφαρμογές πολυμέσων (multimedia)

- Προγραμματίζει δικτυακές εφαρμογές: η Java βοηθάει να υλοποιούνται δικτυακές (network) και διαδικτυακές (web) εφαρμογές.

- Υποστήριξη πολυνηματικής επεξεργασίας (multi-threaded processing): Είναι μία εκ των ελαχίστων γλωσσών στην κατηγορία της υποστηρίζει την δυνατότητα να αναπτυχθούν multi-threaded εφαρμογές.

- Διαχειρίζεται τη μνήμη αυτόματα: Στη Java ο έλεγχος της διαχείρισης ως προς τη μνήμη γίνεται από την ίδια με ένα υποπρόγραμμα που έχει την ονομασία garbage collector (αποκομιστής απορριμμάτων) κι έτσι πραγματοποιείται απεμπλοκή του προγραμματιστή για την συγκεκριμένη ενέργεια.

- Έχει δυναμικότητα: Έχει εύκολη προσαρμογή σε διαφορετικό περιβάλλον και διαφορετικές απαιτήσεις και αποτελεί την ιδανικότερη μορφή γλώσσας για να διασυνδέονται και να επικοινωνούν ετερογενή συστήματα. Υφίσταται συνεχής ενημέρωση της με την ενσωμάτωση και υποστήριξη των τελευταίων τεχνολογικών εξελίξεων [15].

- Χρησιμοποιεί αποκλειστικά την δυναμική διασύνδεση (dynamic binding): Τα δεδομένα που διασυνδέονται και οι μέθοδοι που υποστηρίζονται από αυτά πραγματοποιούνται κατά την διάρκεια που εκτελείται το πρόγραμμα (run-time) [18].

Η πεποίθηση της η οποία αποτελεί τη βάση της φιλοσοφίας της, είναι ότι η γλώσσα επιβάλλεται να προστατεύει τους προγραμματιστές και να υπάρχει ασφάλεια και αξιοπιστία των προγραμμάτων [15].

Η άντληση της συσσωρευμένης γνώσης και εμπειρίας για τον προγραμματισμό στα συστήματα και η ενσωμάτωση των σωστών πρακτικών στον προγραμματισμό είναι οι λόγοι που η Java έχει το χαρακτήρα της πιο ολοκληρωμένης και καλά δομημένης γλώσσας που υπάρχει στην αγορά. Εξ' άλλου δεν είναι θέμα τύχης το γεγονός της ευρείας αποδοχής της άμεσα από τη στιγμή που λανσαρίστηκε καθώς και το ότι οι περισσότεροι επιχειρηματικοί σχεδιασμοί παγκόσμια πραγματοποιούνται με Java [21]. Τρωτό σημείο της όμως είναι η ταχύτητα παρόλο που έχει βελτιωθεί και η οποία παραδόξως οφείλεται στη φορητότητα, το πλέον ισχυρό χαρακτηριστικό της Java. Εξαιτίας της μικρής ταχύτητας δεν χρησιμοποιείται από ένα σύνολο εφαρμογών που είναι σημαντική για να εκτελεστούν π.χ βιντεοπαιχνίδια κ.λ.π. Η παραγωγή από την Java ενδιάμεσου κώδικα που λαμβάνει την ερμηνεία ως κώδικας μηχανής και όχι απ' ευθείας κώδικας μηχανής όπως η C++ προσφέρει την φορητότητα αλλά με το μειονέκτημα της μικρής ταχύτητας [20].

## 2.7 MySQL



Εικόνα 2.2 Το λογότυπο της MySQL.

Η MySQL είναι μια μορφή ελεύθερου συστήματος που διαχειρίζεται βάσεις δεδομένων που έχει διαδοθεί ευρέως στα συστήματα UNIX ωστόσο τελευταία κατέκτησε και τα Win32. Όσοι ενδιαφέρονται έχουν την δυνατότητα εύρεσης στο [www.mysql.com](http://www.mysql.com). Πρόκειται περί μιας πολυνηματικής καθώς και πολλαπλών

χρήσεων γλώσσα υποστηρίζοντας τα πρόσφατα standards της SQL. Εδώ δεν θα ασχοληθούμε με λεπτομέρειες στην πλήρη περιγραφή της αφού γι αυτό θα απαιτηθεί ένας μεγάλος αριθμός σελίδων. Θα δοθεί όμως το σύνολο των βασικών της αρχών και των σημαντικότερων εντολών για τη γρήγορη και εύκολη δημιουργία απλών αλλά και λειτουργικών βάσεων δεδομένων. Το σύνολο των πειραμάτων που θα παρουσιασθούν πιο κάτω πραγματοποιήθηκαν σε RedHat Linux 7.2 και 7.3 που κάνει χρήση της έκδοσης 3.23.41 της MySQL [5].

Η MySQL είναι μια από τις προγραμματιστικές γλώσσες για Συστήματα που διαχειρίζονται Βάσεις Δεδομένων (DBMS). Για να υπάρχει η δυνατότητα κοινών αξόνων αναφοράς των DBMS υπήρξε η ανάγκη δημιουργίας της γλώσσας SQL (Structured Query Language) μέσω της οποίας επιτρέπεται χρησιμοποιώντας εντολές, να προσπελαστούν τα δομημένα δεδομένα που εμπεριέχονται σε ένα DBMS. Επί της ουσίας τα DBMS δέχονται και εκτελούν ένα πακέτο από εντολές της SQL ενώ το MySQL DBMS είναι αποδέκτης εντολών της MySQL [16].

Η MySQL είναι είδος ελεύθερου λογισμικού χρησιμοποιείται από ερασιτέχνες και από επαγγελματίες. Κατ'αντιστοιχία με την MySQL υφίσταται η ύπαρξη κι άλλων δωρεάν ή επί πληρωμή συστημάτων όπως ο SQL Server της Microsoft, ο Oracle SQL server, η DB2 κ.ά.

Με την εξέλιξη της τεχνολογίας των DBMS αυτά διαχωρίστηκαν σύμφωνα με τα δεδομένα που αποθηκεύουν. Μέσω των οργανωμένων δομών δεδομένων πραγματοποιείται αποθήκευση συνόλων δεδομένων (πλειάδες) και του τρόπου βάσει του οποίου έχουν σχέση. Εφόσον η έννοια της λογικής που υφίσταται για να είναι διαχειρίσιμα τα δεδομένα, είναι της συσχέτισης (σχέσης) η ονομασία των βάσεων αυτών είναι Σχισιακές Βάσεις Δεδομένων και των συστημάτων τους σχεσιακά (RDBMS - Relational Database Managements Systems).

Οι βάσεις δεδομένων σε MySQL είναι χώροι κατάλληλα διαμορφωμένοι για να αποθηκευτεί το σύνολο των δεδομένων που περιέχουν οι ιστοσελίδες μας με σχεσιακή μέθοδο [19].

Π.χ αν η ιστοσελίδα μας περιέχει φόρμα για παραγγελία αγαθών τότε η βάση μπορεί να περιέχει τον πίνακα (πλειάδα) με τους χρήστες όπου θα γίνεται η αποθήκευση των πελατών και τον πίνακα (πλειάδα) παραγγελίας που θα γίνεται η αποθήκευση των παραγγελιών τους οι οποίοι θα σχετίζονται με τη σχέση 1 προς πολλά (1:N) αφού υπάρχει το ενδεχόμενο να πραγματοποιηθεί από τον



Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

πελάτη ένας μεγαλύτερος αριθμός παραγγελιών. Μέσω του παραδείγματος αυτού γίνεται αντιληπτός ο τρόπος χρήσης της MySQL στην καθημερινή πρακτική σε θέματα που αφορούν τις ιστοσελίδες [19].

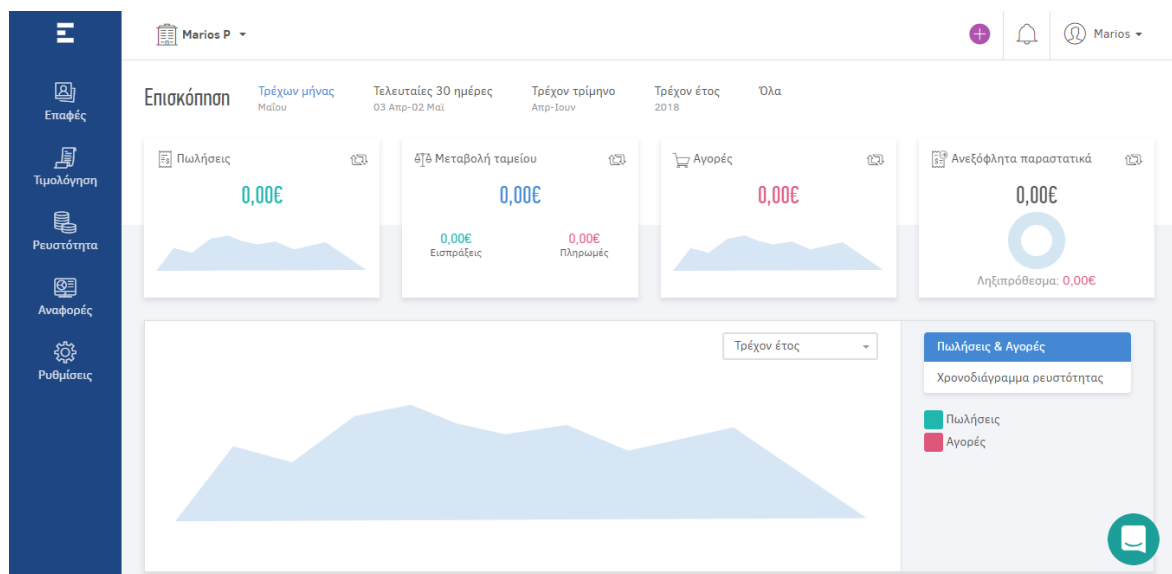


## ΚΕΦΑΛΑΙΟ 3

### ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

#### 3.1 Παρόμοιες εφαρμογές

Για τον σχεδιασμό της εφαρμογής έγινε μελέτη και ανάλυση του τρόπου λειτουργίας άλλων παρόμοιων εφαρμογών. Μια τέτοια είναι η εμπορική διαδικτυακή εφαρμογή τιμολόγησης της elorus (<https://www.elorus.com/>) προσφέροντας σύγχρονες μεθόδους τιμολόγησης, αλλά και επιλογές για να γίνει όσο το δυνατόν πιο προσωποποιημένο για την κάθε επιχείρηση μεμονωμένα.



Εικόνα 3.1 Κεντρική Οθόνη εφαρμογής Elorus

Με την χρήση της κατανοήθηκε ο τρόπος που λειτουργούν οι εφαρμογές τιμολόγησης δίνοντας βασικές πληροφορίες για το πως θα πρέπει να αναπτυχθεί η εφαρμογή και η βάση δεδομένων μου όπως :

- Ποιες είναι οι πρωταρχικές λειτουργίες της εφαρμογής
- Τι δεδομένα είναι απαραίτητο να αποθηκεύονται και για ποιο λόγο
- Την σχέση που υπάρχουν μεταξύ αυτών των δεδομένων
- Το γραφικό περιβάλλον του χρήστη

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

Μαριος P

Επαφές

Προσθήκη Προσθήκη από ΑΦΜ Μαζική εισαγωγή Αναφορές Αποθήκευση

Φίλτρα OFF Ενέργειες Στήλες Αναζήτηση...

Βρέθηκαν 2 επαφές 100 ανά σελί...

Επαφή	ΑΦΜ	Πιστωτικό υπόλοιπο	Ανεξόφλητα παραστατικά	Διευθύνσεις email
<input type="checkbox"/> Μαριος Πατσος - ΜΑΡΙΟΣ ΠΑΤΣΗΣ	146112958	0.00€	0.00€	<input type="button" value="C"/>
<input type="checkbox"/> ΠΑΤΣΗΣ ΠΑΝΑΓΙΩΤΗΣ ΧΡΗΣΤΟΣ	EL146112958	0.00€	0.00€	<input type="button" value="C"/> <input type="button" value="S"/>

Εικόνα 3.2 Οθόνη πελατολογίου

Μαριος P

Προϊόντα / Υπηρεσίες

Προσθήκη Μαζική εισαγωγή Αναφορές Αποθήκευση

Φίλτρα OFF Ενέργειες Στήλες Αναζήτηση...

Βρέθηκε 1 προϊόν / υπηρεσία 100 ανά σελί...

Προϊόν / Υπηρεσία	Κωδικός προϊόντος	Τιμή πώλησης	Πωλήσεις	Τιμή αγοράς	Αγορές	Απόθεμα	Διαχ. αποθέματος
<input type="checkbox"/> aaa		0.00€	<input type="button" value="ΝΑΙ"/>	0.00€	<input type="button" value="ΟΧΙ"/>	0.00	<input type="button" value="ΝΑΙ"/>

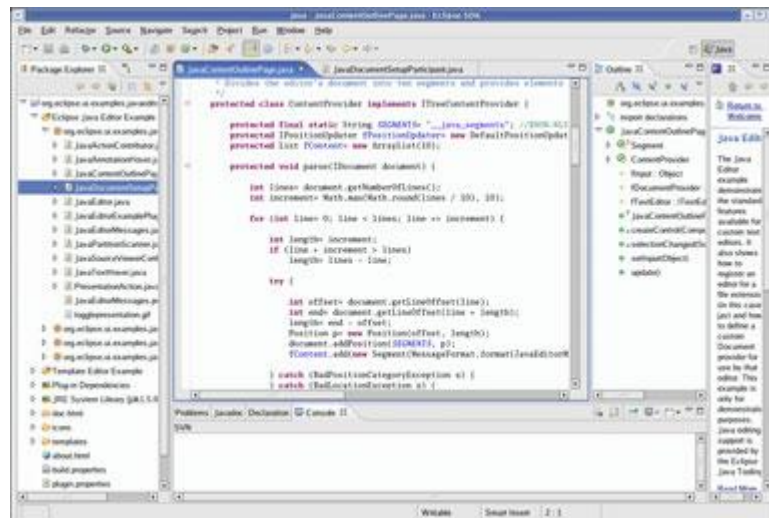
Εικόνα 3.3 Οθόνη Προϊόντα

## 3.2 Εργαλεία ανάπτυξης εφαρμογής

### 3.2.1 Το Eclipse IDE

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) που βοηθάει στην πιο αποδοτική, γρήγορη και ευκολότερη ανάπτυξη προγραμμάτων για υπολογιστές. Το Eclipse όπως και πολλά IDE περιλαμβάνει τις βασικές λειτουργίες που είναι :

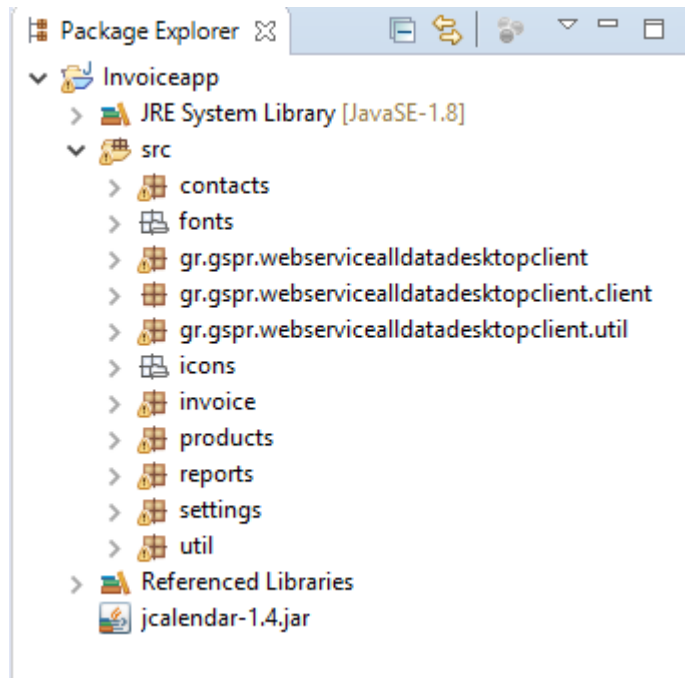
- Επεξεργαστή πηγαίου κώδικα
- Μεταγλωττιστή και διερμηνέα
- Εργαλεία αποσφαλμάτωσης
- Εργαλεία κατασκευής γραφικού περιβάλλοντος χρήστη
- Σύστημα ελέγχου εκδόσεων



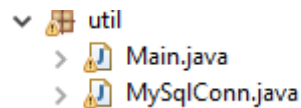
Εικόνα 2.4 Το γραφικό περιβάλλον χρήστη του Eclipse IDE

### 3.2.2 Δομή κώδικα εφαρμογής

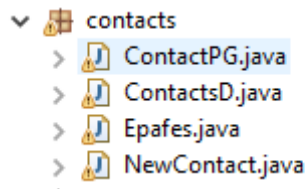
Γίνεται μια συνοπτική αναφορά στα αρχεία κώδικα τα οποία αποτελείται η εφαρμογή μέσα από το περιβάλλον του IDE Eclipse, όπως φαίνεται στις επόμενες εικόνες.



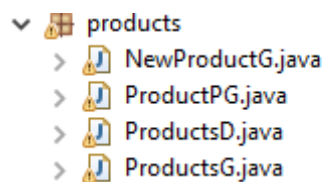
Εικόνα 3.5 Δομή εφαρμογής



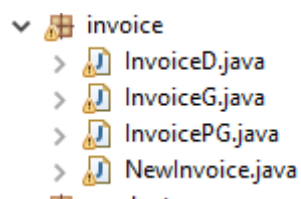
Εικόνα 3.6 Κλάσεις για αρχική οθόνη και σύνδεση με βάση δεδομένων



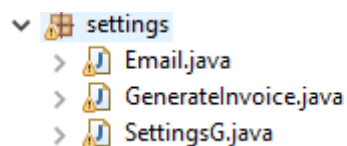
Εικόνα 3.7 Κλάσεις διαχείρισης επαφών



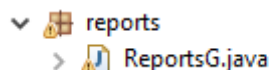
Εικόνα 3.8 Κλάσεις διαχείρισης προϊόντων



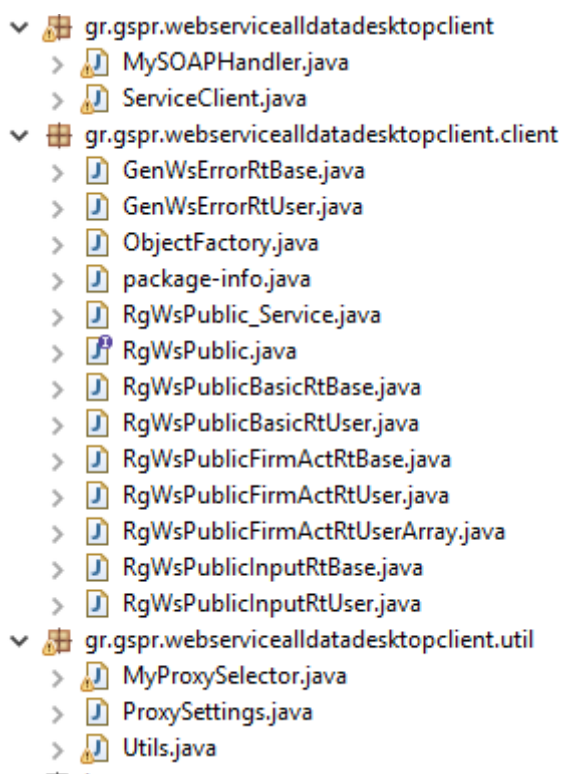
Εικόνα 3.9 Κλάσεις διαχείρισης τιμολογίων



Εικόνα 3.10 Κλάσεις διαχείρισης ρυθμίσεων εφαρμογής και παραγωγής PDF τιμολογίου



Εικόνα 3.11 Κλάσεις για αναφορές

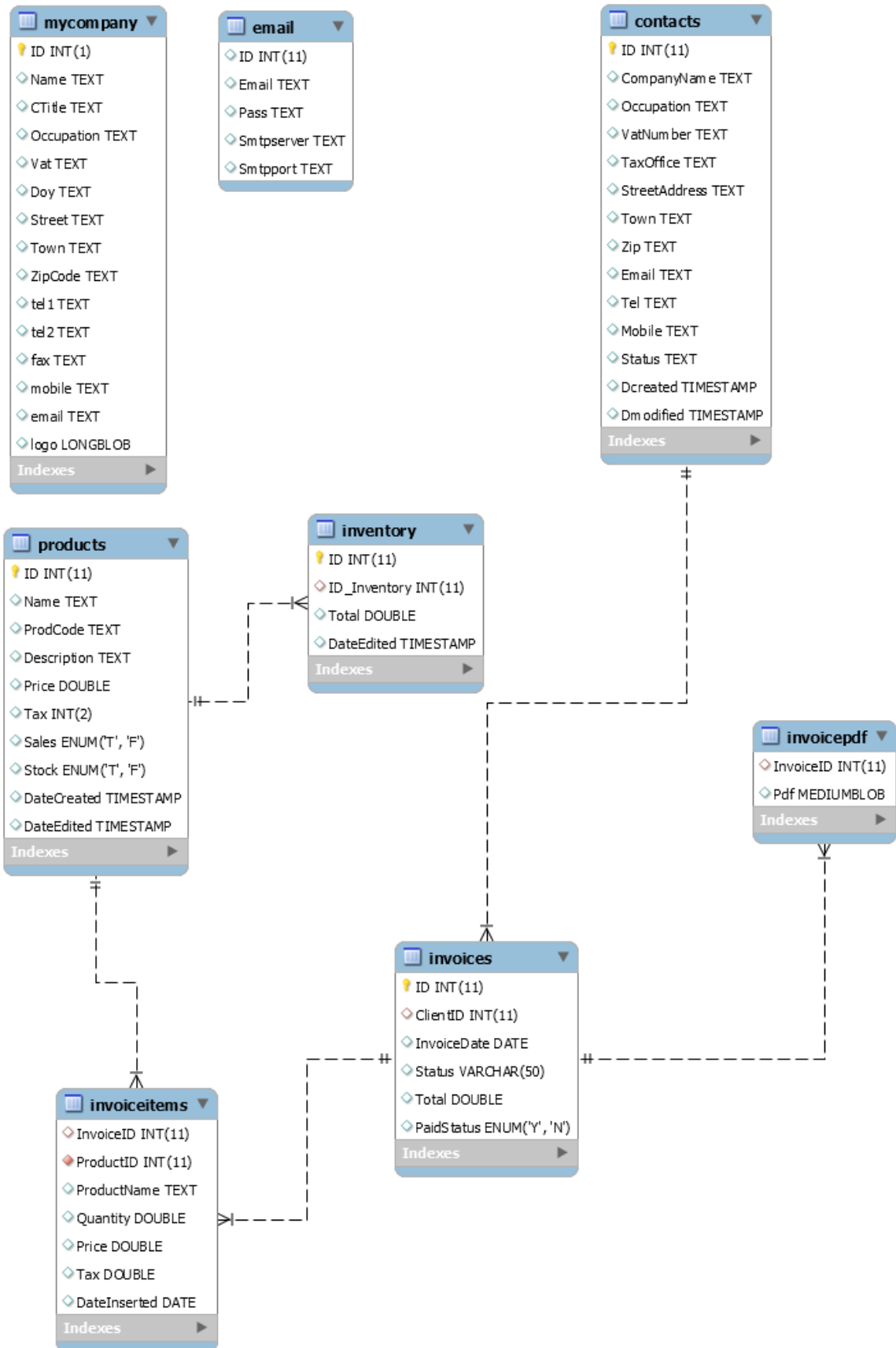


Εικόνα 3.12 Κλάσεις σύνδεσης εφαρμογής με ΓΓΠΣ

### 3.2.3 Τα δεδομένα της εφαρμογής

Για όλα τα δεδομένα της εφαρμογής χρησιμοποιήθηκε βάση δεδομένων και πιο συγκεκριμένα η MySQL. Με αυτόν τον τρόπο μπορεί να γίνει οργανωμένη αποθήκευση και εξαγωγή πληροφορίας μέσα από ένα μεγάλο όγκο δεδομένων και σχέσεων που έχουμε στην εφαρμογή. Εκτός από την οργανωμένη δομή των δεδομένων με τη χρήση βάσης δεδομένων δίνουμε την δυνατότητα να χρησιμοποιηθεί η εφαρμογή από πολλούς χρήστες ταυτόχρονα και τα δεδομένα κάθε στιγμή να είναι διαθέσιμα σε όλους.

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση



Εικόνα 3.13 Σχηματικό βάσης δεδομένων εφαρμογής



### **3.2.4 Δημιουργία εγγράφων PDF**

Η εφαρμογή κατά την έκδοση ενός παραστατικού εξάγει όλα τα απαραίτητα δεδομένα (στοιχεία πελάτη, εμπόρου, προϊόντα, σύνολα κ.α.) σε ένα έγγραφο pdf έτσι ώστε να είναι ευανάγνωστο από ένα άτομο που θα το διαβάσει αλλά και να έχει μια όμορφη παρουσίαση.

Για την δημιουργία εγγράφων σε μορφή PDF χρησιμοποιείται η βιβλιοθήκη iText (<https://itextpdf.com/>) όπου παρέχει τη δυνατότητα να δημιουργήσουμε μέσω κώδικα Java έγγραφα PDF.

### **3.2.5 Υπηρεσία ΓΓΠΣ για αναζήτηση – έλεγχο ΑΦΜ**

Η εφαρμογή για την αναζήτηση στοιχείων και χρήση αυτών για την δημιουργία μιας επαφής αυτόματα μέσω ενός αριθμού ΑΦΜ χρησιμοποιεί την υπηρεσία που προσφέρει η ΓΓΠΣ. Έτσι μπορεί να αντλεί γρηγορά και έγκυρα στοιχεία για ένα ΑΦΜ το μόνο που προϋποθέτει είναι :

- Να κάνει εγγραφή στην διαδικτυακή υπηρεσία «Βασικά στοιχεία για νομικά πρόσωπα, νομικές οντότητες και φυσικά πρόσωπα με εισόδημα από επιχειρηματική δραστηριότητα» χρησιμοποιώντας τους Taxisnet κωδικούς του
- Αποδοχή της πολιτικής ορθής χρήσης
- Να αποκτήσει ειδικούς κωδικούς μέσω της εφαρμογής «Διαχείριση Ειδικών Κωδικών» του Taxisnet[26]

Πηγαίος κώδικας client εφαρμογής που χρησιμοποιείται στην εφαρμογή υπάρχει στην ιστοσελίδα της ΓΓΠΣ

([http://www.gsis.gr/gsis/info/gsis\\_site/PublicIssue/wnsp/wnsp\\_pages/webserviceclient.html](http://www.gsis.gr/gsis/info/gsis_site/PublicIssue/wnsp/wnsp_pages/webserviceclient.html))



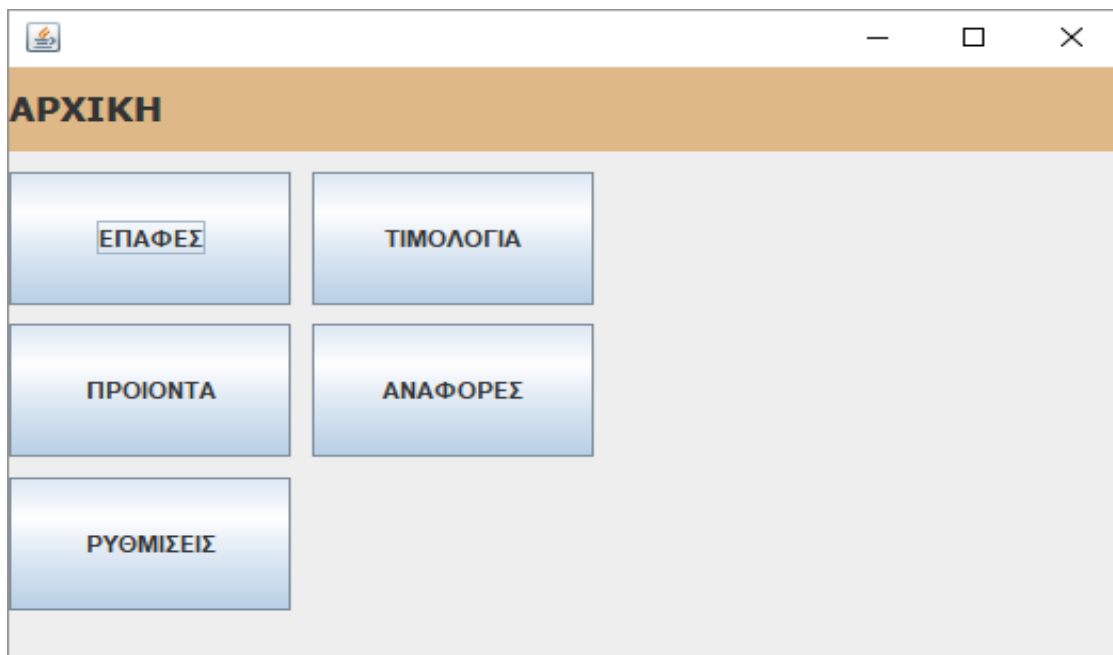
## ΚΕΦΑΛΑΙΟ 4

### ΕΡΕΥΝΗΤΙΚΟ ΚΟΜΜΑΤΙ

Στο παρόν κεφάλαιο παρουσιάζεται η εφαρμογή που δημιουργήθηκε και αναλύεται ο τρόπος λειτουργίας της και οι ιδιότητές της, μέσω εικόνων και επεξηγήσεων.

#### 4.1 Αρχική οθόνη

Αφού εκτελέσουμε την εφαρμογή, μεταφερόμαστε στην αρχική οθόνη μέσω της οποίας ο χρήστης μπορεί να επιλέξει ανάμεσα από τις πέντε βασικές λειτουργίες που προσφέρει η εφαρμογή.



Εικόνα 4.1 Αρχική οθόνη

#### 4.2 Ρυθμίσεις

Πατώντας το κουμπί ρυθμίσεις στην αρχική οθόνη μεταφερόμαστε στην οθόνη ρυθμίσεις όπου υπάρχουν δυο κουμπιά, το ένα είναι στοιχεία επιχείρησης όπου γίνεται εισαγωγή των βασικών στοιχείων της επιχείρησης και το άλλο email για την εισαγωγή στοιχείων του λογαριασμού ηλεκτρονικού ταχυδρομείου (email).

ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

EMAIL

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

ΔΙΑΚΡΙΤΟΣ ΤΙΤΛΟΣ

ΔΡΑΣΤΗΡΙΟΤΗΤΑ

ΑΦΜ

ΔΟΥ

ΔΙΕΥΘΥΝΣΗ

ΠΕΡΙΟΧΗ

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΟΣ

ΤΗΛΕΦΩΝΟ 1

ΚΙΝΗΤΟ

email

ΤΗΛΕΦΩΝΟ 2

Fax

ΑΝΑΖΗΤΗΣΗ

ΔΙΑΓΡΑΦΗ

ΕΠΙΣΤΡΟΦΗ

ΕΦΑΡΜΟΓΗ

Εικόνα 4.2 οθόνη αλλαγής στοιχείων επιχείρησης

#### 4.2.1 Στοιχεία επιχείρησης

Στην παρακάτω εικόνα (εικόνα 4.3) παρουσιάζεται η διαδικασία εισαγωγής των στοιχείων της επιχείρησης που θα εμφανιστούν στο τελικό τιμολόγιο.

ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

EMAIL

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

ΠΑΤΣΗΣ ΜΑΡΙΟΣ

ΔΙΑΚΡΙΤΟΣ ΤΙΤΛΟΣ

ΠΑΤΣΗΣ

ΔΡΑΣΤΗΡΙΟΤΗΤΑ

ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΦΜ

146112306

ΔΟΥ

Ν.ΙΩΝΙΑΣ

ΔΙΕΥΘΥΝΣΗ

Κ.ΒΑΡΝΑΛΗ 1

ΠΕΡΙΟΧΗ

ΑΘΗΝΑ

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΟΣ

14231

ΤΗΛΕΦΩΝΟ 1

698047726

ΚΙΝΗΤΟ

6980477426

email

mariospatsis@gmail.com

ΤΗΛΕΦΩΝΟ 2

Fax

6980477426

ΑΝΑΖΗΤΗΣΗ

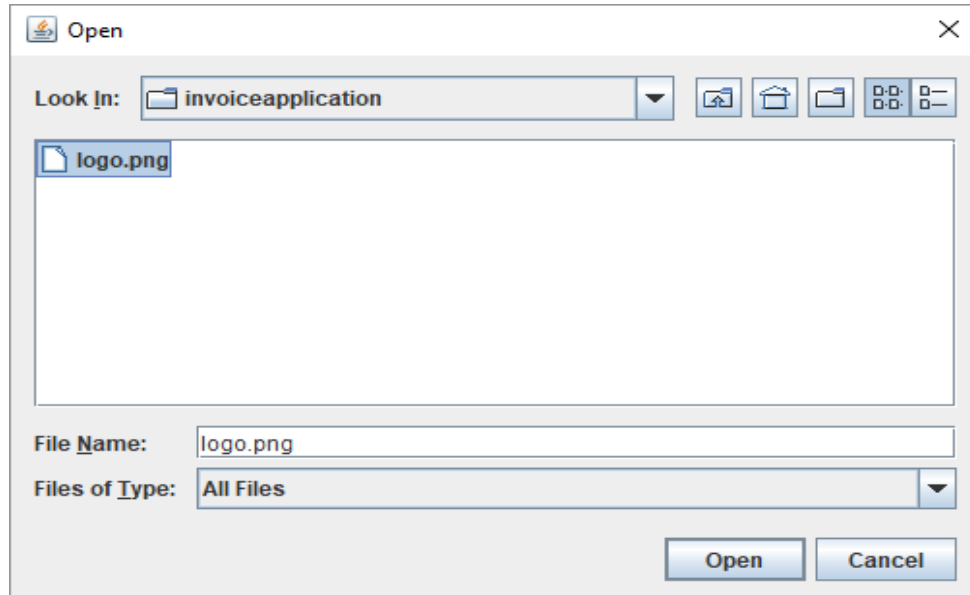
ΔΙΑΓΡΑΦΗ

ΕΠΙΣΤΡΟΦΗ

ΕΦΑΡΜΟΓΗ

Εικόνα 4.3 Συμπληρωμένη οθόνη εισαγωγής στοιχείων επιχείρησης

Για την εισαγωγή του λογότυπου πατάμε το κουμπί αναζήτηση για να ανοίξει το παράθυρο διαλόγου ώστε να βρούμε μέσα από τον υπολογιστή την εικόνα που θέλουμε. (εικόνα 4.4)



Εικόνα 4.4 Παράθυρο επιλογής εικόνας

Στην συνέχεια πατάμε open(άνοιγμα) για επιλογή εικόνας ή cancel (ακύρωση).

Το «Files of Type» υπάρχει ως επιλογή για να εμφανίζει μόνο τα αρχεία εικόνας στους φακέλους που κάνουμε αναζήτηση.

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗΣΗΣ	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	ΔΙΑΚΡΙΤΟΣ ΤΙΤΛΟΣ	ΠΑΤΣΗΣ
ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ		
ΑΦΜ	146112306	ΔΟΥ	Ν.ΙΩΝΙΑΣ
ΔΙΕΥΘΥΝΣΗ	Κ.ΒΑΡΝΑΛΗ 1	ΠΕΡΙΟΧΗ	ΑΘΗΝΑ
		ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΟΣ	14231
ΤΗΛΕΦΩΝΟ 1	698047726	ΚΙΝΗΤΟ	6980477426
		email	mariospatsis@gmail.com
ΤΗΛΕΦΩΝΟ 2		Fax	6980477426

ΑΝΑΖΗΤΗΣΗ  
ΔΙΑΓΡΑΦΗ

ΕΠΙΣΤΡΟΦΗ ΕΦΑΡΜΟΓΗ

PATSIIS

Εικόνα 4.5 Συμπληρωμένη οθόνη εισαγωγής στοιχείων επιχείρησης.

Με το κουμπί διαγραφή, διαγράφεται η εικόνα που επιλέξαμε ή που υπάρχει ήδη στην βάση δεδομένων σαν επιλεγμένη εικόνα .

Με το κουμπί εφαρμογή αποθηκεύονται όλες οι αλλαγές και κλείνει το παράθυρο επιστρέφοντας τον χρήστη στην αρχική οθόνη.

Το κουμπί επιστροφή δεν αποθηκεύει καμία αλλαγή, κλείνει το παράθυρο ρυθμίσεις και επιστρέφει στην αρχική οθόνη.

#### 4.2.2 Email

Το κουμπί Email στις ρυθμίσεις, εμφανίζει την οθόνη για την επεξεργασία των ρυθμίσεων ηλεκτρονικού ταχυδρομείου (email) του χρήστη, δηλαδή ο χρήστης πρέπει να συμπληρώσει τα παρακάτω για να γίνει χρήση της αποστολής ηλεκτρονικού ταχυδρομείου μέσω της εφαρμογής :

- Email : συμπληρώνεται η διεύθυνση ηλεκτρονικού ταχυδρομείου του αποστολέα
- Κωδικός : ο κωδικός της διεύθυνσης ηλεκτρονικού ταχυδρομείου

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

- emailSMTPserver : Η διεύθυνση smtp του server του ηλεκτρονικού ταχυδρομείου του αποστολέα
- emailSMTPserverport: Η πόρτα για τον συγκεκριμένο server

ΑΛΛΓΗ ΡΥΘΜΙΣΕΩΝ EMAIL

ΣΤΟΙΧΕΙΑ ΕΠΙΧΕΙΡΗΣΗΣ

EMAIL

email : invoiceapptest@gmail.com

κωδικός : testtestinvoice

emailSMTPserv... smtp.gmail.com

emailSMTPserv... 465

ΕΦΑΡΜΟΓΗ ΑΚΥΡΩΣΗ

Εικόνα 4.6 Οθόνη αλλαγής ρυθμίσεων email.

### 4.3 Επαφές

ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ
----	----------	-----

Εικόνα 4.7 Οθόνη επαφές

Με το κουμπί νέα επαφή δημιουργούμε μια καινούργια επαφή. Το πλαίσιο δεξιά είναι πεδίο αναζήτησης, κάνει αναζήτηση με βάση την επωνυμία ή με βάση το ΑΦΜ της επαφής. Η αναζήτηση γίνεται κατά την πληκτρολόγηση και δεν χρειάζεται να πατηθεί άλλο πλήκτρο.(εικόνα 4.8)



The screenshot shows a web application window with the title "ΕΠΑΦΕΣ". At the top left, there is a button labeled "ΝΕΑ ΕΠΑΦΗ". To its right is a search input field containing the text "ΠΑΤ". Below the search bar is a table with three columns: "ID", "ΕΠΩΝΥΜΙΑ", and "ΑΦΜ". The table contains one row of data. At the bottom of the window, there are two buttons: "ΕΜΦΑΝΙΣΗ ΕΠΑΦΗΣ" on the left and "ΕΠΙΣΤΡΟΦΗ" on the right.

ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015

Εικόνα 4.8 Αναζήτηση επαφής

#### 4.3.1 Εμφάνιση επαφής

Για να ανοίξουμε την καρτέλα της επαφής και να δούμε όλα τα στοιχεία της επιλέγουμε την επαφή και πατάμε το κουμπί εμφάνιση επαφής. (εικόνα 4.9)

ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015

Εικόνα 4.9 Επιλογή επαφής για εμφάνιση

Στην καρτέλα επαφής εμφανίζονται τα εξής στοιχεία στο αριστερό μέρος της οθόνης :

- Επωνυμία
- Δραστηριότητα
- ΑΦΜ
- ΔΟΥ
- Διεύθυνση
- Πόλη
- Ταχυδρομικός κώδικας
- Email
- Τηλέφωνο

- Κινητο
- Ημερομηνία δημιουργίας επαφής
- Ημερομηνία τροποποίησης επαφής

και τέλος δεξιά στην οθόνη εμφανίζεται πίνακας με τα ανεξόφλητα παραστατικά της επαφής αυτής. (εικόνα 4.10)

ΜΗ ΕΞΟΦΛΗΜΕΝΑ ΤΙΜΟΛΟΓΙΑ		
ID	ΗΜ.ΕΚΔΟΣΗΣ	ΠΟΣΟ
1	2018-04-10	496.0

Εικόνα 4.10 Καρτέλα επαφής

#### 4.3.2 Τροποποίηση στοιχείων

Με το κουμπί διόρθωση στοιχείων μπορεί να γίνει αλλαγή στοιχείων της επαφής όπως φαίνεται παρακάτω στην εικόνα (εικόνα 4.11). Σε περίπτωση που γίνει κάποια τροποποίηση της επαφής αλλάζει και η ημερομηνία τροποποίησης. Παρακάτω έχει αλλαχθεί η διεύθυνση της επαφής αλλάζοντας και την ημερομηνία τροποποίησης(εικόνα 4.12).

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

**ΕΠΕΞΕΡΓΑΣΙΑ ΕΠΑΦΗΣ**

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ... \*  
ΠΑΤΣΗΣ ΜΑΡΙΟΣ

EMAIL  
mariospatsis@gmail.com

ΔΡΑΣΤΗΡΙΟΤΗΤΑ \*  
ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΗΛΕΦΩΝΟ  
6980477426

ΑΦΜ \* ΔΟΥ \*  
146113015 ΝΕΑΣ ΙΩΝΙΑΣ

ΚΙΝΗΤΟ  
6980477426

ΔΙΕΥΘΥΝΣΗ \* ΠΟΛΗ \*  
Κ.ΒΑΡΝΑΛΗ 2 ΑΘΗΝΑ

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ...  
14231

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.11 Αλλαγή στοιχείων επαφής

**ΕΠΑΦΗ**

ΕΠΩΝΥΜΙΑ : ΠΑΤΣΗΣ ΜΑΡΙΟΣ

ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΦΜ: 146113015

ΔΟΥ: ΝΕΑΣ ΙΩΝΙΑΣ

ΔΙΕΥΘΥΝΣΗ : Κ.ΒΑΡΝΑΛΗ 2

ΠΟΛΗ : ΑΘΗΝΑ

ΤΑΧ.Κ: 14231

email : mariospatsis@gmail.com

ΤΗΛ : 6980477426

ΚΙΝ : 6980477426

Ημ.Τροποποίησης 2018-04-10 18:28:49.0

Ημ. Δημιουργίας 2018-04-10 17:28:09.0

**ΜΗ ΕΞΟΦΛΗΜΕΝΑ ΤΙΜΟΛΟΓΙΑ**

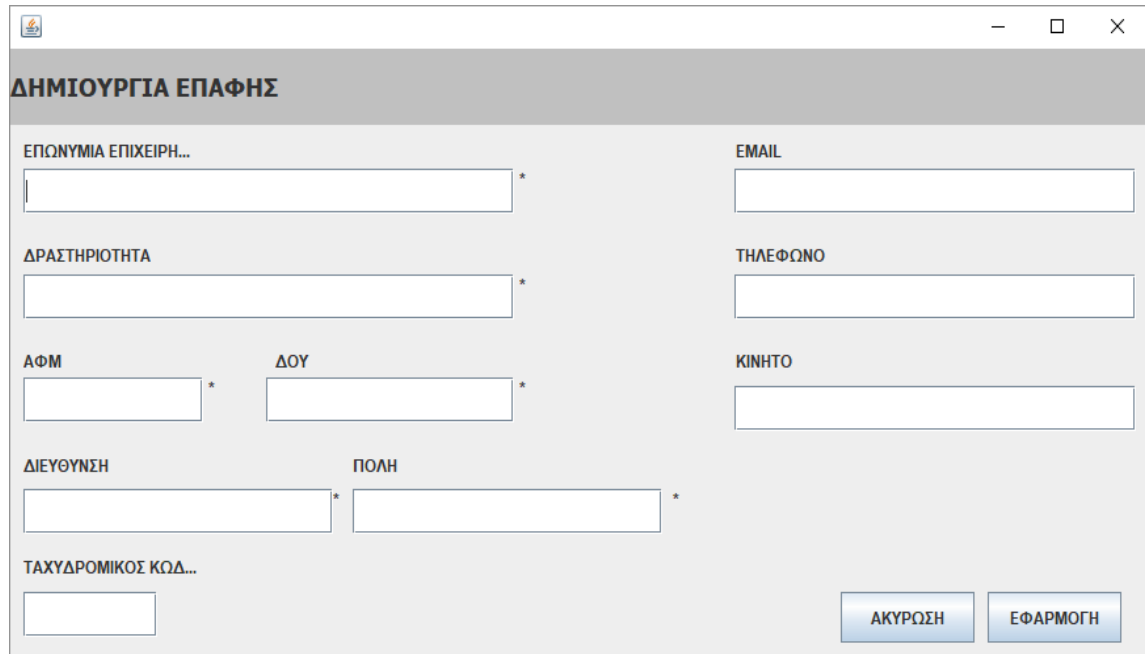
ID	ΗΜ.ΕΚΔΟΣΗΣ	ΠΟΣΟ
1	2018-04-10	496.0

ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙΩΝ ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.12 Αλλαγή ημερομηνίας τροποποίησης μετά από αλλαγή στοιχείων επαφής.

### 4.3.3 Νέα επαφή

Με το κουμπί νέα επαφή φτιάχνουμε μια καινούργια επαφή. (εικόνα 4.13)



ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ... \*

EMAIL

ΔΡΑΣΤΗΡΙΟΤΗΤΑ \*

ΤΗΛΕΦΩΝΟ

ΑΦΜ \* ΔΟΥ \*

ΚΙΝΗΤΟ

ΔΙΕΥΘΥΝΣΗ \* ΠΟΛΗ \*

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ...

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.13 Δημιουργία επαφής

Όπου υπάρχει αστερίσκος (\*) είναι πεδίο που πρέπει να συμπληρωθεί για να υπάρξει επιτυχής δημιουργία επαφής. Μπορεί να γίνει εισαγωγή στοιχείων με δυο τρόπους:

1. Συμπληρώνοντας ένα – ένα τα κελιά χειροκίνητα.
2. Μέσω της διαδικτυακής εφαρμογής της Γενικής Γραμματείας Πληροφοριακών Συστημάτων όπου αντλεί βασικά στοιχεία για νομικά πρόσωπα, νομικές οντότητες και φυσικά πρόσωπα που έχουν εισόδημα από επιχειρηματική δραστηριότητα. Έτσι η συμπλήρωση των περισσότερων κελίων που απαιτούνται γίνεται αυτόματα απλα συμπληρώνοντας το ΑΦΜ του πελάτη στο αντίστοιχο κελί.

**ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ**

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ... \*

EMAIL

ΔΡΑΣΤΗΡΙΟΤΗΤΑ \*

ΤΗΛΕΦΩΝΟ

ΑΦΜ \* ΔΟΥ \*

ΚΙΝΗΤΟ

ΔΙΕΥΘΥΝΣΗ \* ΠΟΛΗ \*

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ...

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.14 Συμπλήρωση Α.Φ.Μ για αναζήτηση

Όταν συμπληρωθούν όλα τα ψηφία του ΑΦΜ ελέγχεται μέσω της διαδικτυακής υπηρεσίας άμα το ΑΦΜ είναι έγκυρο και βρέθηκαν στοιχεία στην υπηρεσία της Γενικής Γραμματείας Πληροφοριακών Συστημάτων για το συγκεκριμένο ΑΦΜ . Σε αυτήν την περίπτωση εμφανίζεται το αντίστοιχο πράσινο βελάκι. Αμα δεν βρεθεί το ΑΦΜ τότε δεν εμφανίζεται πράσινο βελάκι. (εικόνα 4.15)

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

The screenshot shows a web application window titled "ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ". The form contains several input fields with asterisks indicating they are required. The "ΑΦΜ" field contains the value "146112958" and has a green checkmark next to it. The "ΔΟΥ" field is empty. The "EMAIL", "ΤΗΛΕΦΩΝΟ", and "ΚΙΝΗΤΟ" fields are empty. The "ΔΙΕΥΘΥΝΣΗ" and "ΠΟΛΗ" fields are empty. The "ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ..." field is empty. At the bottom right, there are two buttons: "ΑΚΥΡΩΣΗ" and "ΕΦΑΡΜΟΓΗ".

Εικόνα 4.15 Επιτυχής αναζήτηση ΑΦΜ

Όταν πατηθεί το πράσινο βελάκι γίνεται συμπλήρωση των παρακάτω κελιών όπως φαίνεται στην παρακάτω εικόνα εκτός της δραστηριότητας, του email, του τηλεφώνου και του κινητού τηλεφώνου.(εικόνα 4.16)

The screenshot shows the same "ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ" form, but now with data entered in several fields. The "ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ..." field contains "ΠΑΤΣΗΣ ΠΑΝΑΓΙΩΤΗΣ ΧΡΗΣΤΟΣ". The "ΑΦΜ" field contains "146112958" and has a green checkmark. The "ΔΟΥ" field contains "ΝΕΑΣ ΙΩΝΙΑΣ". The "ΔΙΕΥΘΥΝΣΗ" field contains "ΠΑΡΗΣΗΣ11". The "ΠΟΛΗ" field contains "Ν ΙΩΝΙΑ". The "ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ..." field contains "14231". The "EMAIL", "ΤΗΛΕΦΩΝΟ", and "ΚΙΝΗΤΟ" fields are still empty. The "ΑΚΥΡΩΣΗ" and "ΕΦΑΡΜΟΓΗ" buttons are still present at the bottom right.

Εικόνα 4.16 Αυτόματη συμπλήρωση κελιών

Σε περίπτωση που πατηθεί το κουμπί εφαρμογή χωρίς να έχουν συμπληρωθεί όλα τα απαιτούμενα πεδία όπως παραπάνω που δεν έχει συμπληρωθεί το απαιτούμενο πεδίο της δραστηριότητα τότε εμφανίζεται το παρακάτω μήνυμα το οποίο ενημερώνει το χρήστη αντίστοιχα. (εικόνα 4.17)

ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ... ΠΑΤΣΗΣ ΠΑΝΑΓΙΩΤΗΣ ΧΡΗΣΤΟΣ \*

EMAIL

ΔΡΑΣΤΗΡΙΟΤΗΤΑ \*

ΤΗΛΕΦΩΝΟ

ΑΦΜ 146112958 \* ΔΟΥ ΝΕΑΣ ΙΩΝΙΑΣ \*

ΔΙΕΥΘΥΝΣΗ ΠΑΡΣΗΣ11 \* ΠΟΛΗ Ν ΙΩΝΙΑ \*

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ... 14231

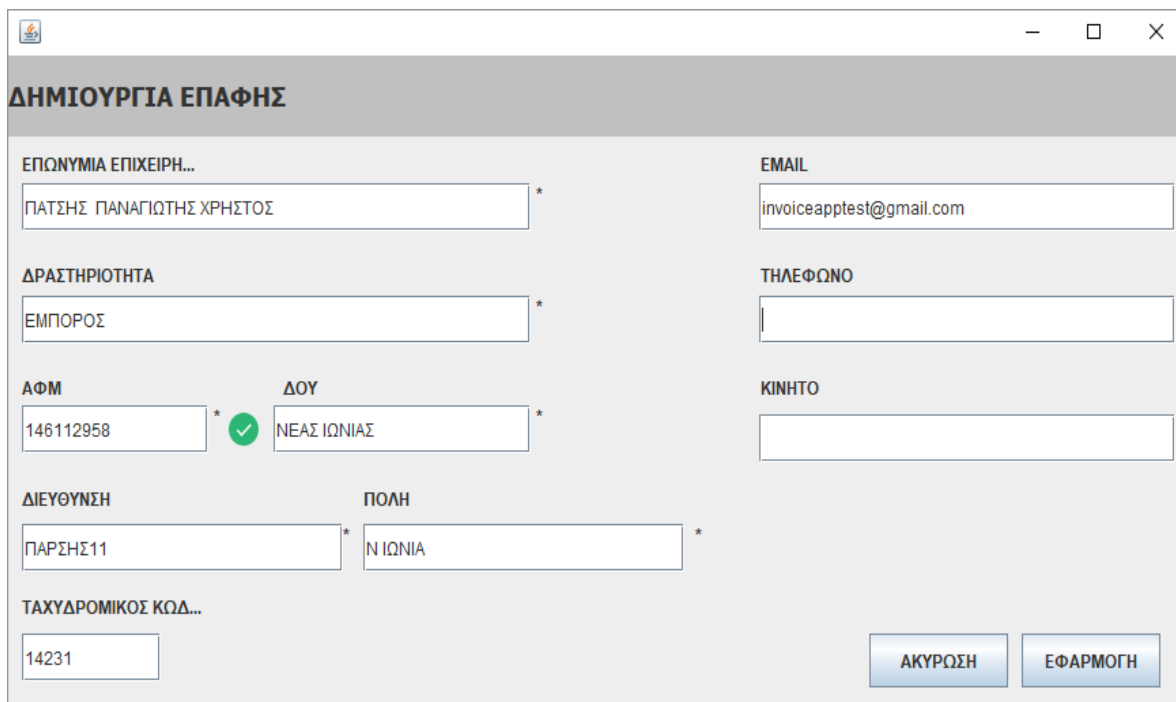
ΠΡΟΣΟΧΗ  
ΕΙΣΑΓΕΤΕ ΟΛΑ ΤΑ ΑΠΑΙΤΟΥΜΕΝΑ ΚΕΛΙΑ  
OK

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.17 Μήνυμα προς χρήστη

Ο χρήστης θα πρέπει να συμπληρώσει όλα τα απαιτούμενα κελιά και να πατήσει το κουμπί εφαρμογή για να γίνει αποθήκευση της καινούργιας επαφής. (εικόνα 4.18)





**ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ**

ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗ...  
ΠΑΤΣΗΣ ΠΑΝΑΓΙΩΤΗΣ ΧΡΗΣΤΟΣ \*

EMAIL  
invoiceapptest@gmail.com

ΔΡΑΣΤΗΡΙΟΤΗΤΑ  
ΕΜΠΟΡΟΣ \*

ΑΦΜ  
146112958 \* ✓

ΔΟΥ  
ΝΕΑΣ ΙΩΝΙΑΣ \*

ΔΙΕΥΘΥΝΣΗ  
ΠΑΡΣΗΣ11 \*

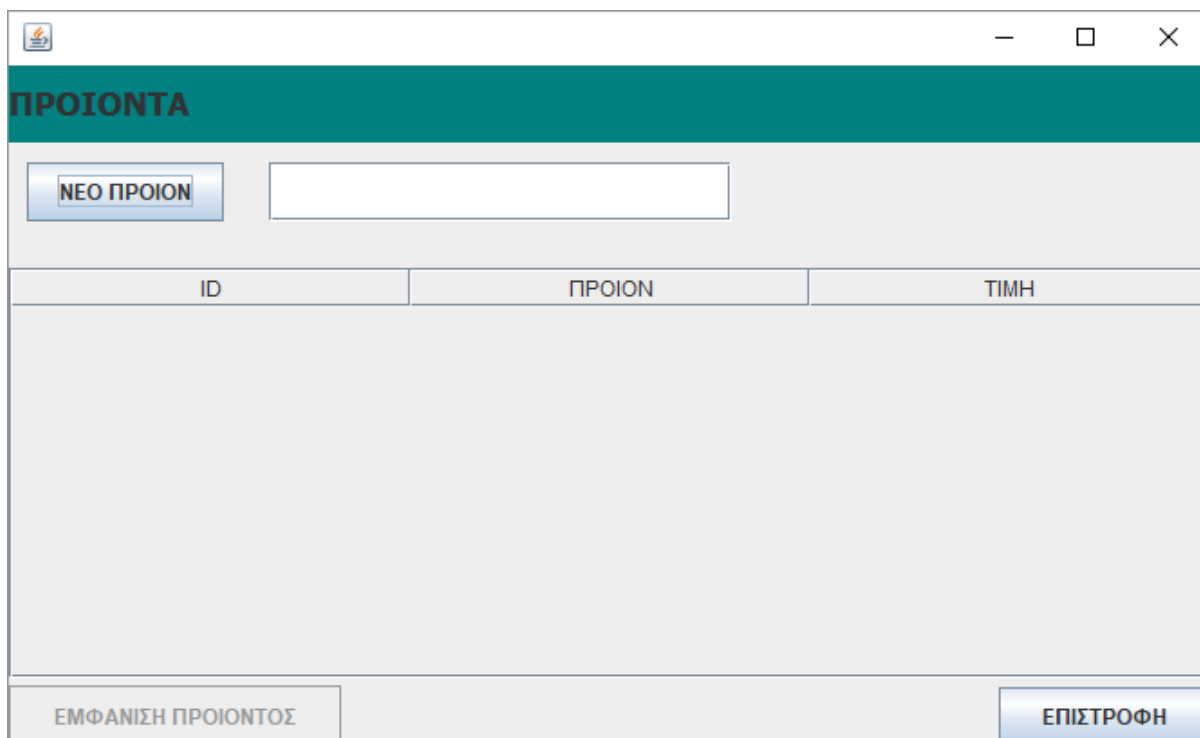
ΠΟΛΗ  
Ν ΙΩΝΙΑ \*

ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔ...  
14231

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.18 Νέα επαφή με συμπληρωμένα όλα τα απαιτούμενα κελιά

#### 4.4 Προϊόντα



**ΠΡΟΙΟΝΤΑ**

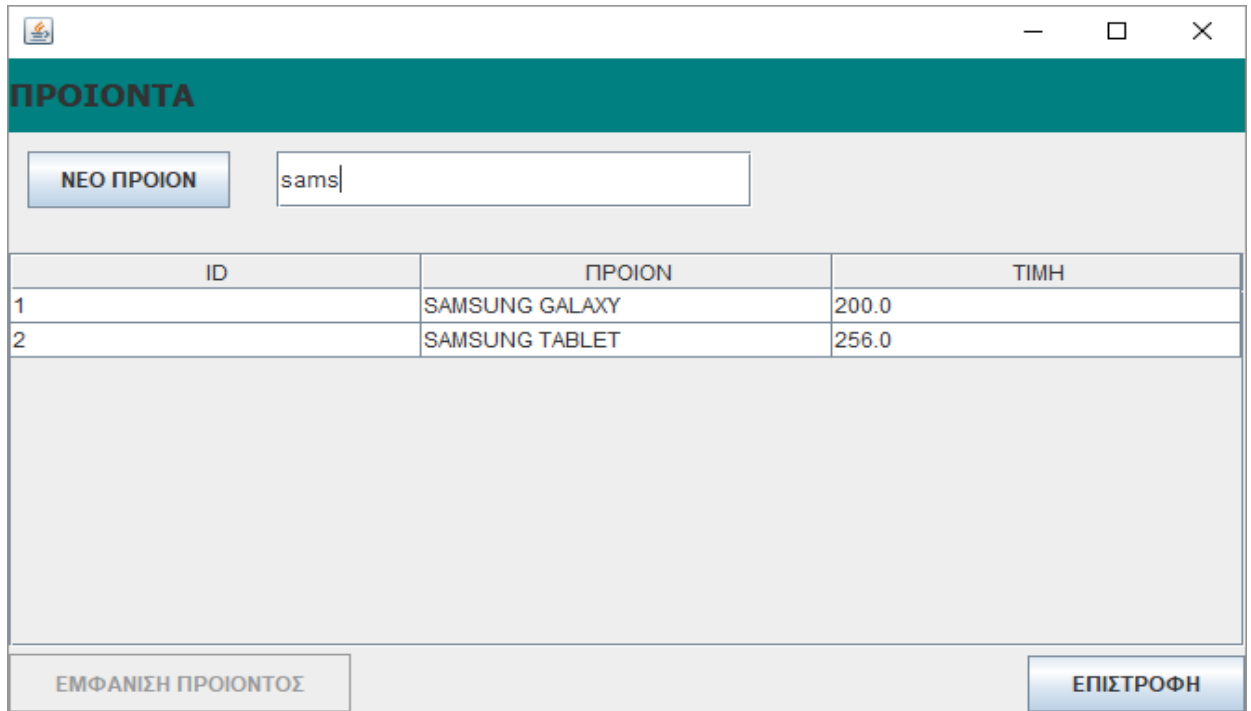
ΝΕΟ ΠΡΟΙΟΝ

ID	ΠΡΟΙΟΝ	ΤΙΜΗ
----	--------	------

ΕΜΦΑΝΙΣΗ ΠΡΟΙΟΝΤΟΣ ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.19 Οθόνη προϊόντα

Με το κουμπί νέο προϊόν δημιουργούμε ένα καινούργιο προϊόν. Το πεδίο δεξιά είναι για αναζήτηση προϊόντων, κάνει αναζήτηση με βάση την ονομασία του προϊόντος ή με βάση τον κωδικό προϊόντος. Η αναζήτηση γίνεται κατά την πληκτρολόγηση και δεν χρειάζεται να πατηθεί άλλο πλήκτρο. (εικόνα 4.20)

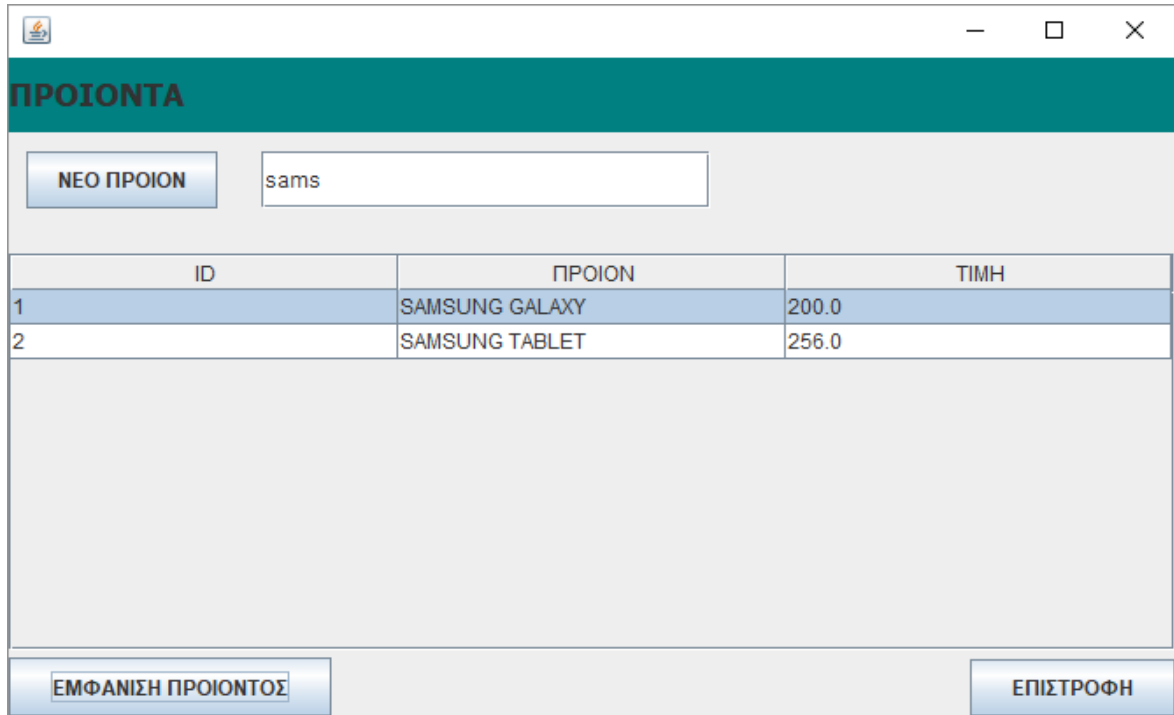


ID	ΠΡΟΙΟΝ	ΤΙΜΗ
1	SAMSUNG GALAXY	200.0
2	SAMSUNG TABLET	256.0

Εικόνα 4.20 Αναζήτηση προϊόντος

#### 4.4.1 Εμφάνιση προϊόντος

Για να εμφανιστούν όλα τα στοιχεία ενός προϊόντος επιλέγουμε το προϊόν και πατάμε το κουμπί εμφάνιση προϊόντος. (εικόνα 4.21)



Εικόνα 4.21 Εμφάνιση προϊόντος

Στην καρτέλα προϊόν εμφανίζονται τα εξής στοιχεία στο αριστερό μέρος :

- Όνομα προϊόντος
- Κωδικός προϊόντος
- Περιγραφή προϊόντος
- Τιμή
- Φ.Π.Α
- Κουμπί Ενεργό – Ανενεργό
- Κουμπί Αποθήκη Ενεργή – Μη ενεργή
- Ημερομηνία δημιουργίας
- Ημερομηνία τροποποίησης

Ενώ στο δεξί μέρος της οθόνης :

- Διαθέσιμη αποθήκη

Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

- Πλαίσιο για εισαγωγή αριθμού ώστε να γίνει προσθήκη αποθήκης ή αφαίρεσή αποθήκης
- Ημερομηνία τελευταίας αλλαγής της αποθήκης

**ΠΡΟΙΟΝ**

ΟΝΟΜΑ ΠΡΟΪΟΝΤΟΣ : SAMSUNG GALAXY  
ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟ... 000152456412  
ΠΕΡΙΓΡΑΦΗ ΠΡΟΪΟΝΤ... ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΔΙΑΘΕΣΙΜΗ ΑΠΟΘΗΚΗ : 23.0

ΤΙΜΗ : 200.0  
Φ.Π.Α : 24

0

ΠΡΟΣΘ... ΑΦΑΙΡΕ...

ΗΜ. ΤΕΛΕΥΤ ΑΛΛΑΓΗ...  
2018-04-10 18:24:32.0

ΕΝΕΡΓΟ

ΑΠΟΘΗΚΗ ΕΝΕΡΓΗ

ΗΜ. ΔΗΜΙΟΥΡΓΙΑΣ : 2018-04-10 18:17:36.0  
ΗΜ. ΤΕΛΕΥΤ ΤΡΟΠΟΠ : 2018-04-10 19:11:50.0

ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙ... ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.22 Καρτέλα προϊόν

Η αποθήκη μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί το πλήκτρο αποθήκη ενεργή – Αποθήκη ανενεργή (εικόνα 4.23)

**ΠΡΟΙΟΝ**

ΟΝΟΜΑ ΠΡΟΪΟΝΤΟΣ : SAMSUNG GALAXY ΔΙΑΘΕΣΙΜΗ ΑΠΟΘΗΚΗ : 23.0

ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟ... 000152456412

ΠΕΡΙΓΡΑΦΗ ΠΡΟΪΟΝΤ... ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΤΙΜΗ : 200.0

Φ.Π.Α : 24

ΠΡΟΣΘ... ΑΦΑΙΡΕ...

ΗΜ. ΤΕΛΕΥΤ ΑΛΛΑΓΗ... 2018-04-10 18:24:32.0

ΕΝΕΡΓΟ

ΑΠΟΘΗΚΗ ΑΝΕΝΕΡ...

ΗΜ. ΔΗΜΙΟΥΡΓΙΑΣ : 2018-04-10 18:17:36.0

ΗΜ. ΤΕΛΕΥΤ ΤΡΟΠΟΠ : 2018-04-10 19:11:50.0

ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙ... ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.23 Αλλαγή κατάστασης αποθήκης

Με τα κουμπιά προσθήκη – αφαίρεσή αλλάζουμε το διαθέσιμο αριθμό προϊόντων στην αποθήκη. Κατά την τροποποίηση της αποθήκης αλλάζει και η ημερομηνία τελευταίας αλλαγής της αποθήκης.

Με το κουμπί ενεργό – ανενεργό αλλάζουμε την κατάσταση του προϊόντος για το αν μπορεί να γίνει χρήση του ή όχι. Δηλαδή όταν είναι ανενεργό ένα προϊόν εμφανίζεται στην αναζήτηση αλλά όταν γίνει επιλογή στην έκδοση τιμολογίου τότε θα εμφανιστεί μήνυμα στον χρήστη ότι δεν μπορεί να γίνει η χρήση του. (εικόνα 4.24)

**ΠΡΟΙΟΝ**

ΟΝΟΜΑ ΠΡΟΪΟΝΤΟΣ : SAMSUNG GALAXY  
ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟ... 000152456412  
ΠΕΡΙΓΡΑΦΗ ΠΡΟΪΟΝΤ... ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΤΙΜΗ : 200.0  
Φ.Π.Α : 24

ΔΙΑΘΕΣΙΜΗ ΑΠΟΘΗΚΗ : 23.0

0

ΠΡΟΣΘ... ΑΦΑΙΡΕ...

ΑΝΕΝΕΡΓΟ

ΑΠΟΘΗΚΗ ΕΝΕΡΓΗ

ΗΜ. ΔΗΜΙΟΥΡΓΙΑΣ : 2018-04-10 18:17:36.0  
ΗΜ. ΤΕΛΕΥΤ ΤΡΟΠΟΠ : 2018-04-10 19:26:42.0

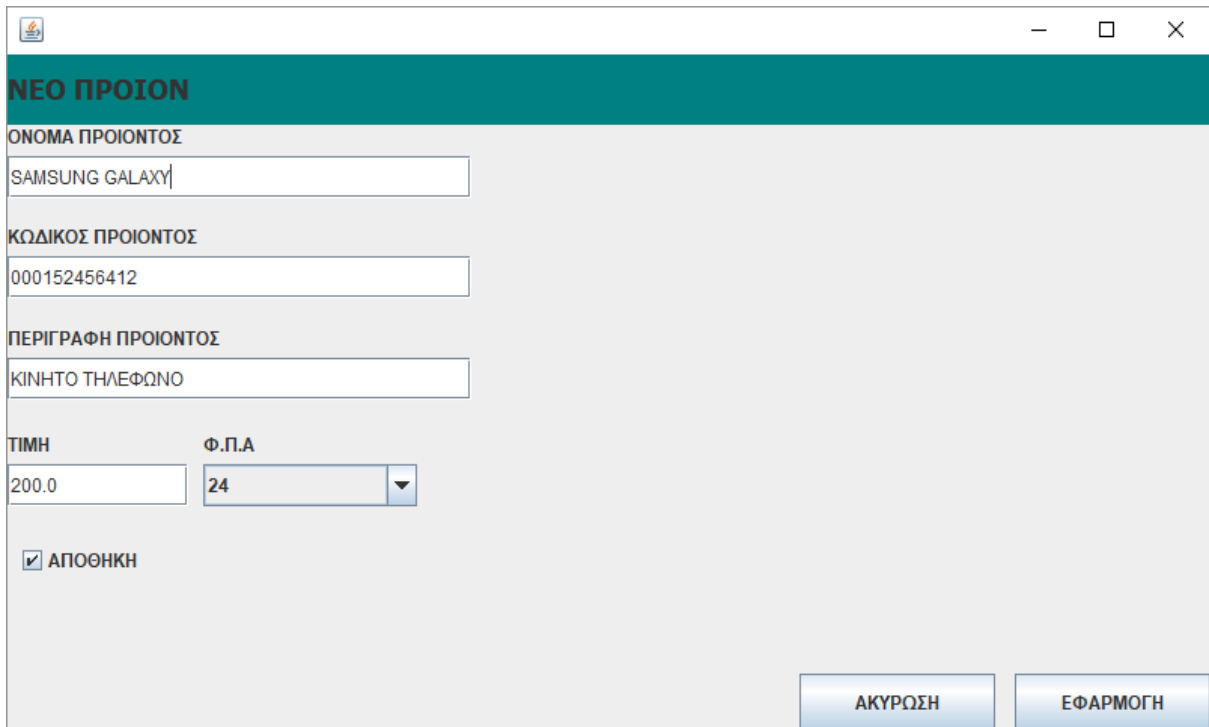
ΗΜ. ΤΕΛΕΥΤ ΑΛΛΑΓΗ...  
2018-04-10 18:24:32.0

ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙ... ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.24 Προϊόν Ενεργό - Ανενεργό

#### 4.4.2 Τροποποίηση προϊόντος

Με το κουμπί διόρθωση στοιχείων γίνεται η τροποποίηση των στοιχείων του προϊόντος. Σε περίπτωση που γίνει κάποια τροποποίηση αλλάζει και η ημερομηνία τροποποίησης.(εικόνα 4.25)



**ΝΕΟ ΠΡΟΙΟΝ**

ΟΝΟΜΑ ΠΡΟΙΟΝΤΟΣ  
SAMSUNG GALAXY

ΚΩΔΙΚΟΣ ΠΡΟΙΟΝΤΟΣ  
000152456412

ΠΕΡΙΓΡΑΦΗ ΠΡΟΙΟΝΤΟΣ  
ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΤΙΜΗ Φ.Π.Α  
200.0 24

ΑΠΟΘΗΚΗ

ΑΚΥΡΩΣΗ ΕΦΑΡΜΟΓΗ

Εικόνα 4.25 Τροποποίηση προϊόντος

#### 4.4.3 Δημιουργία προϊόντος

Με το κουμπί νέο προϊόν στην οθόνη προϊόντα δημιουργούμε ένα καινούργιο προϊόν.

Τα πεδία που πρέπει να συμπληρωθούν είναι :

- Όνομα προϊόντος
- Κωδικός προϊόντος
- Περιγραφή προϊόντος
- Τιμή
- Φ.Π.Α 6% , 13% ή 24%
- Επιλογή ενεργοποίησης αποθήκης

The screenshot shows a web application window titled "NEO ΠΡΟΙΟΝ". It contains several input fields: "ΟΝΟΜΑ ΠΡΟΙΟΝΤΟΣ" (empty), "ΚΩΔΙΚΟΣ ΠΡΟΙΟΝΤΟΣ" (empty), "ΠΕΡΙΓΡΑΦΗ ΠΡΟΙΟΝΤΟΣ" (empty), "ΤΙΜΗ" (empty), and "Φ.Π.Α" (dropdown menu with "24" selected). There is also a checkbox labeled "ΑΠΟΘΗΚΗ" which is unchecked. At the bottom right, there are two buttons: "ΑΚΥΡΩΣΗ" and "ΕΦΑΡΜΟΓΗ".

Εικόνα 4.26 Οθόνη νέο προϊόν

The screenshot shows the same "NEO ΠΡΟΙΟΝ" form, but now with data entered: "ΟΝΟΜΑ ΠΡΟΙΟΝΤΟΣ" contains "APPLE IPHONE", "ΚΩΔΙΚΟΣ ΠΡΟΙΟΝΤΟΣ" contains "145547447", "ΠΕΡΙΓΡΑΦΗ ΠΡΟΙΟΝΤΟΣ" contains "ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ", "ΤΙΜΗ" contains "800", and "Φ.Π.Α" remains "24". The "ΑΠΟΘΗΚΗ" checkbox is still unchecked. The "ΑΚΥΡΩΣΗ" and "ΕΦΑΡΜΟΓΗ" buttons are visible at the bottom right.

Εικόνα 4.27 Συμπλήρωση κελιών για τη δημιουργία καινούργιου προϊόντος



## 4.5 Τιμολόγια

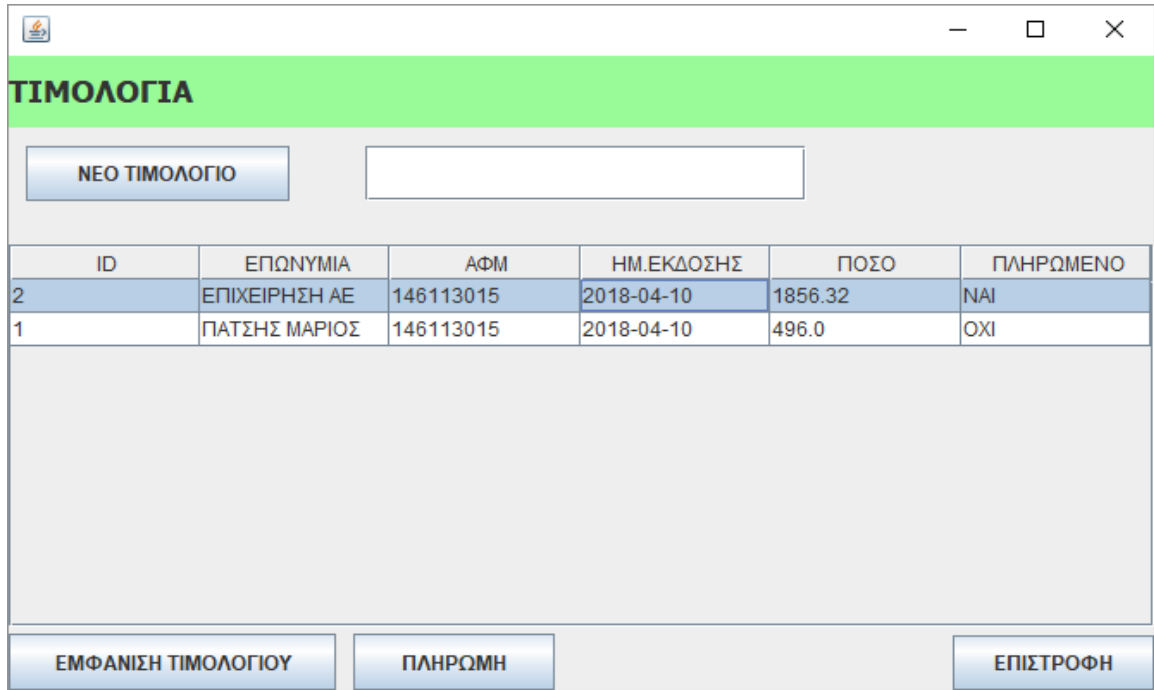
Με κουμπί νέο τιμολόγιο δημιουργούμε ένα καινούργιο τιμολόγιο. Το κελί δίπλα είναι πεδίο αναζήτησης, γίνεται αναζήτηση με βάση την επωνυμία στο οποίο έχει εκδοθεί το τιμολόγιο ή με βάση τον Α.Φ.Μ του πελάτη. Η αναζήτηση γίνεται κατά την πληκτρολόγηση και δεν χρειάζεται να πατηθεί άλλο πλήκτρο. Κατά την είσοδο στην οθόνη τιμολόγια εμφανίζονται τα τελευταία 20 τιμολόγια που έχουν εκδοθεί. (εικόνα 4.28)

ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ	ΗΜ.ΕΚΔΟΣΗΣ	ΠΟΣΟ	ΠΛΗΡΩΜΕΝΟ
2	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	146113015	2018-04-10	1856.32	ΟΧΙ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015	2018-04-10	496.0	ΟΧΙ

Εικόνα 4.28 Οθόνη τιμολόγια

### 4.5.1 Πληρωμή

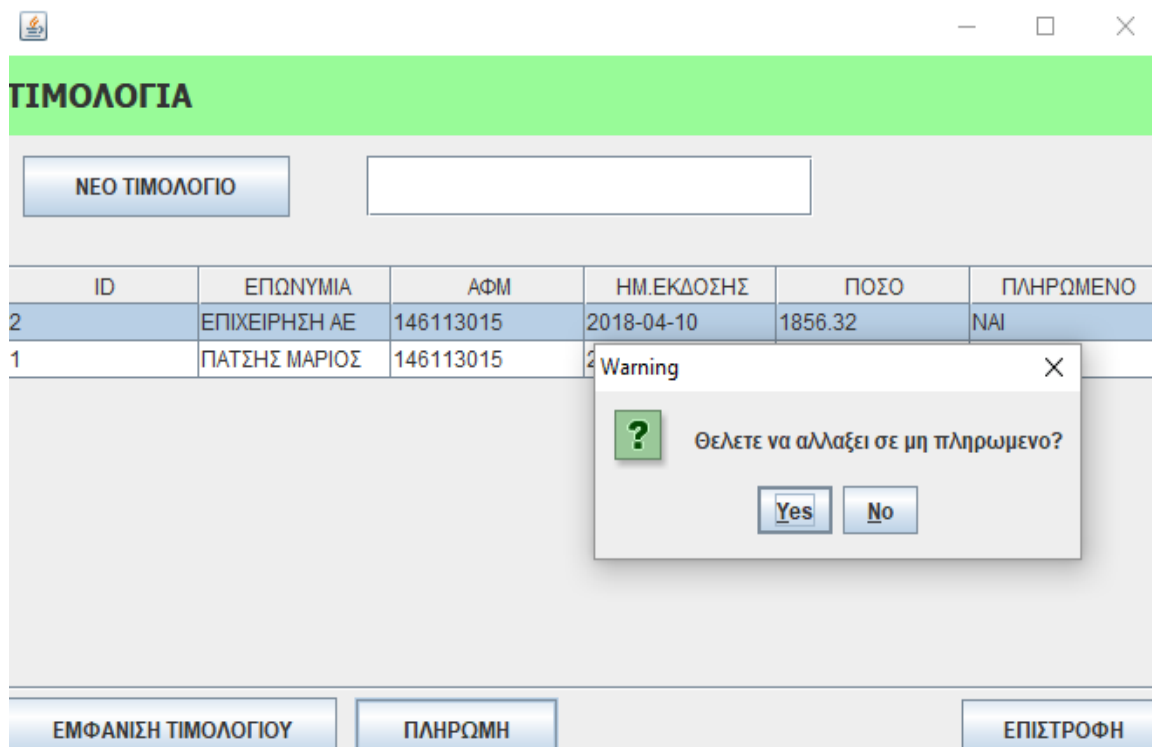
Για να γίνει αλλαγή ενός τιμολογίου σε πληρωμένο (δηλαδή να εμφανίζεται στο κελί του πίνακα πληρωμένο με ένδειξη ναι) επιλέγουμε το τιμολόγιο και πατάμε το κουμπί πληρωμή. Το κελί από όχι αλλάζει σε ναι.



ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ	ΗΜ.ΕΚΔΟΣΗΣ	ΠΟΣΟ	ΠΛΗΡΩΜΕΝΟ
2	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	146113015	2018-04-10	1856.32	ΝΑΙ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015	2018-04-10	496.0	ΟΧΙ

Εικόνα 4.29 Πληρωμή τιμολογίου

Για να αλλάξουμε την κατάσταση του τιμολογίου από πληρωμένο σε απλήρωτο τότε επιλέγουμε το τιμολόγιο και πατάμε το πλήκτρο πληρωμή, όπου εμφανίζεται το μήνυμα ενημερώνοντας τον χρήστη ότι η κατάσταση του τιμολογίου θα αλλάξει σε μη πληρωμένο. (εικόνα 4.30)



Εικόνα 4.30 Αλλαγή σε μη πληρωμένο

Επιλέγουμε το τιμολόγιο και πατάμε το κουμπί εμφάνιση τιμολογίου για να μεταφερθούμε στην οθόνη τιμολόγιο όπου εμφανίζονται στοιχεία για το παραστατικό που έχει αποθηκευτεί (εικόνα 4.31)

ΤΙΜΟΛΟΓΙΟ																
ΕΠΩΝΥΜΙΑ : ΕΠΙΧΕΙΡΗΣΗ ΑΕ		ΑΦΜ : 146113015														
ΗΜ.ΕΚΔΟΣΗΣ : 2018-04-10		ΑΡ.ΠΑΡΑΣΤΑΤΙΚ... 2														
ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ												
APPLE IPHONE	1.0	800.0	13.0	800.0												
SAMSUNG TABLET	3.0	256.0	24.0	768.0												
<table style="width: 100%; border: none;"> <tr> <td style="border: none;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">ΑΠΟΘΗΚΕΥΣΗ</td> <td style="width: 50%; text-align: center;">ΑΠΟΣΤΟΛΗ</td> </tr> <tr> <td style="width: 50%; text-align: center;">ΑΝΟΙΓΜΑ</td> <td style="width: 50%;"></td> </tr> </table> </td> <td style="border: none; vertical-align: top;">                 ΦΠΑ 24 : 184.32                  ΦΠΑ 13 : 104.0                  ΦΠΑ 6 :             </td> <td style="border: none; vertical-align: top;">                 ΣΥΝ. ΠΛΗΝ ΦΠΑ : 1568.0                  ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 1856.32             </td> </tr> <tr> <td style="border: none; text-align: center;">ΕΠΙΣΤΡΟΦΗ</td> <td colspan="4" style="border: none;"></td> </tr> </table>					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">ΑΠΟΘΗΚΕΥΣΗ</td> <td style="width: 50%; text-align: center;">ΑΠΟΣΤΟΛΗ</td> </tr> <tr> <td style="width: 50%; text-align: center;">ΑΝΟΙΓΜΑ</td> <td style="width: 50%;"></td> </tr> </table>	ΑΠΟΘΗΚΕΥΣΗ	ΑΠΟΣΤΟΛΗ	ΑΝΟΙΓΜΑ		ΦΠΑ 24 : 184.32 ΦΠΑ 13 : 104.0 ΦΠΑ 6 :	ΣΥΝ. ΠΛΗΝ ΦΠΑ : 1568.0 ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 1856.32	ΕΠΙΣΤΡΟΦΗ				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">ΑΠΟΘΗΚΕΥΣΗ</td> <td style="width: 50%; text-align: center;">ΑΠΟΣΤΟΛΗ</td> </tr> <tr> <td style="width: 50%; text-align: center;">ΑΝΟΙΓΜΑ</td> <td style="width: 50%;"></td> </tr> </table>	ΑΠΟΘΗΚΕΥΣΗ	ΑΠΟΣΤΟΛΗ	ΑΝΟΙΓΜΑ		ΦΠΑ 24 : 184.32 ΦΠΑ 13 : 104.0 ΦΠΑ 6 :	ΣΥΝ. ΠΛΗΝ ΦΠΑ : 1568.0 ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 1856.32										
ΑΠΟΘΗΚΕΥΣΗ	ΑΠΟΣΤΟΛΗ															
ΑΝΟΙΓΜΑ																
ΕΠΙΣΤΡΟΦΗ																

Εικόνα 4.31 Οθόνη τιμολόγιο

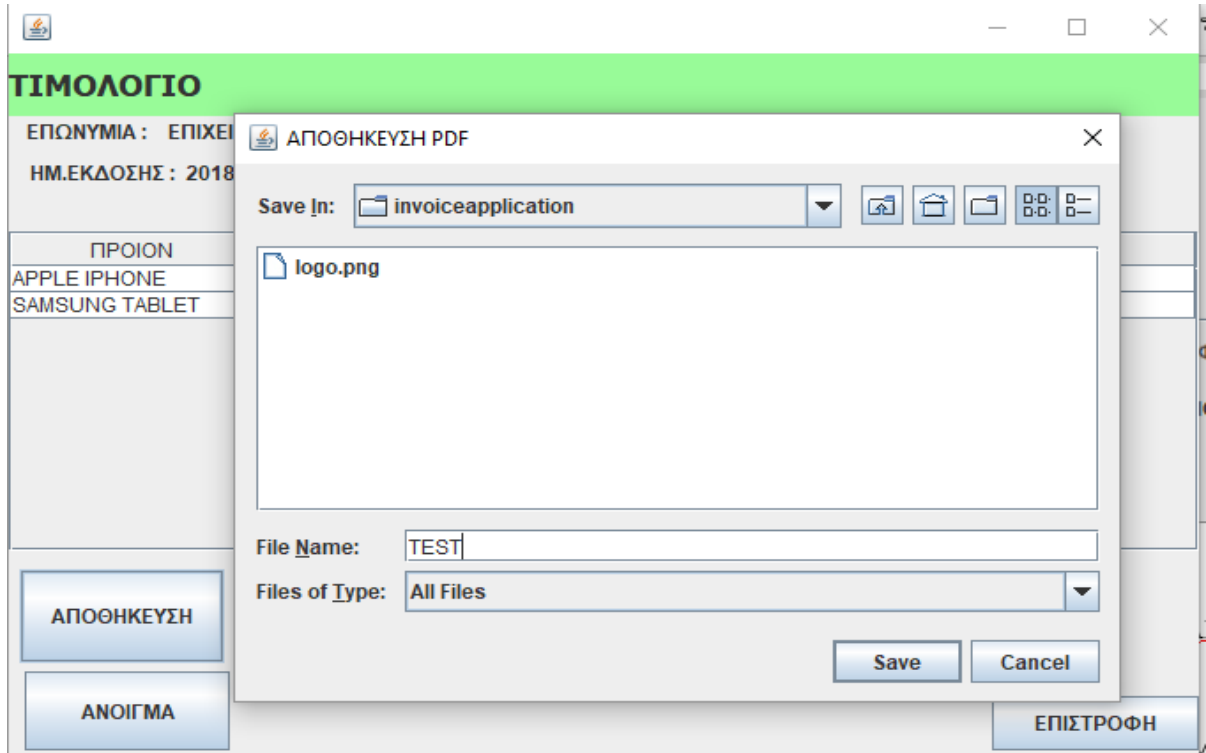
Ο χρήστης έχει τις εξής επιλογές πατώντας τα αντίστοιχα κουμπιά :

Άνοιγμα : Γίνεται εμφάνιση του παραστατικού με το προεπιλεγμένο πρόγραμμα εκτέλεσης αρχείων σε μορφή PDF του λειτουργικού συστήματος. (εικόνα 4.36)

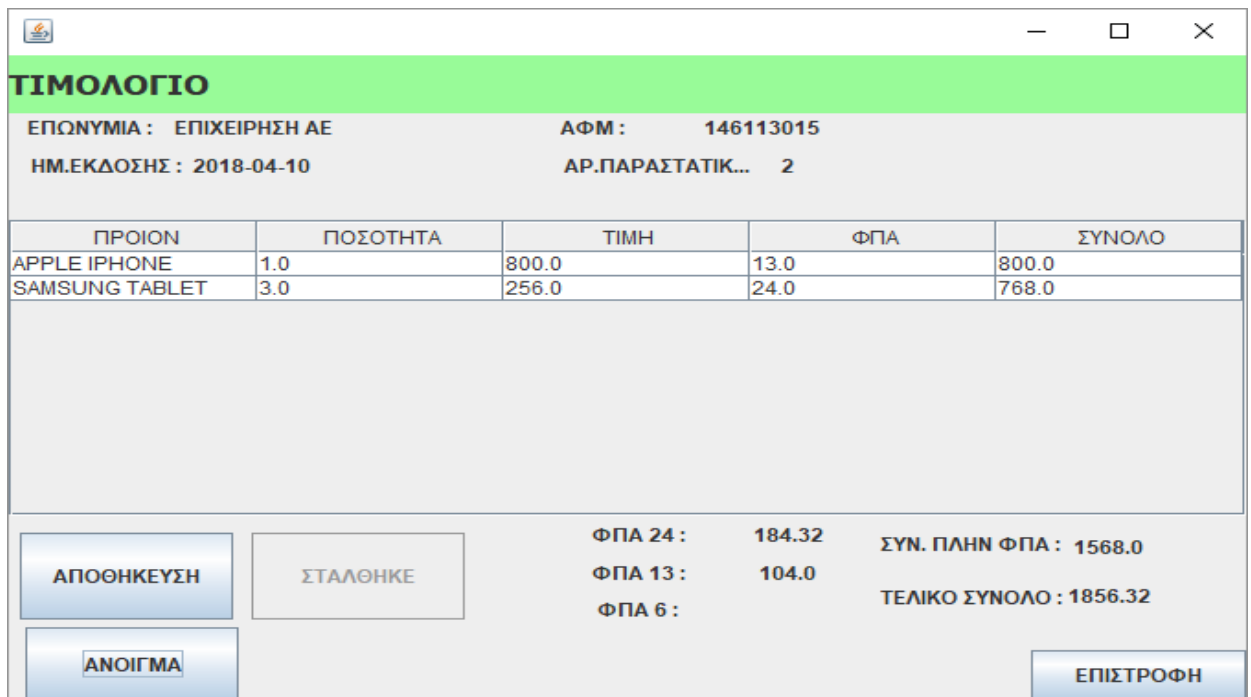
Αποθήκευση : Γίνεται εξαγωγή του παραστατικού σε αρχείο PDF και αποθηκεύεται στην τοποθεσία που θα του ορίσει ο χρήστης. (εικόνα 4.32)

Αποστολή : Γίνεται αποστολή του παραστατικού σε μορφή PDF μέσω ηλεκτρονικού ταχυδρομείου, στην διεύθυνση ηλεκτρονικού ταχυδρομείου (email) που υπάρχει στην επαφή του πελάτη. Στην επιτυχή αποστολή το κουμπί αποστολή γίνεται στάλθηκε ενώ σε περίπτωση που δεν υπάρχει διεύθυνση ταχυδρομείου (email) στην επαφή του πελάτη εμφανίζεται μήνυμα στο χρήστη. (εικόνες 4.33 & 4.34 & 4.35)

Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση



Εικόνα 4.32 Αποθήκευση σε αρχείο



Εικόνα 4.33 Αποστολή σε email

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση


**ΤΙΜΟΛΟΓΙΟ**

ΕΠΩΝΥΜΙΑ : ΕΠΙΧΕΙΡΗΣΗ ΑΕ  
 ΗΜ.ΕΚΔΟΣΗΣ : 2018-04-10

ΑΦΜ : 146113015  
 ΑΡ.ΠΑΡΑΣΤΑΤΙΚ... 2

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ
APPLE IPHONE	1.0	800.0	13.0	800.0
SAMSUNG TABLET	3.0	256.0	24.0	768.0

InfoBox: ΠΡΟΣΟΧΗ ×

 ΔΕΝ ΥΠΑΡΧΕΙ ΔΙΕΥΘΥΝΣΗ EMAIL ΣΤΗΝ ΕΠΑΦΗ

OK

ΑΠΟΘΗΚΕΥΣΗ

ΑΠΟΣΤΟΛΗ

ΑΝΟΙΓΜΑ

ΦΠΑ 24 : 184.32      ΣΥΝ. ΠΛΗΝ ΦΠΑ : 1568.0

ΦΠΑ 13 : 104.0        ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 1856.32

ΦΠΑ 6 :

ΕΠΙΣΤΡΟΦΗ

Εικόνα 4.34 Δεν υπάρχει email

Το μήνυμα ηλεκτρονικού ταχυδρομείου με το συνημμένο PDF. (εικόνα 4.35)

Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση



ΤΙΜΟΛΟΓΙΟ Inbox x




invoiceapptest@gmail.com

to me ▾

ΤΙΜΟΛΟΓΙΟ ΜΕ ΑΡΙΘΜΟ ΠΑΡΑΣΤΑΤΙΚΟΥ2



Εικόνα 4.35 Email με συνημμένο



**ΑΠΟ**  
**ΠΑΤΣΗΣ**  
**ΠΑΤΣΗΣ ΜΑΡΙΟΣ**  
**ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΑΦΜ: 146112306 ΔΟΥ: Ν.ΙΩΝΙΑΣ**  
**Κ.ΒΑΡΝΑΛΗ 1**  
**ΑΘΗΝΑ Τ.Κ: 14231**  
**Τηλ1 698047726 Τηλ2**  
**Κιν 6980477426 fax 6980477426**  
**email mariospatsis@gmail.com**

**ΤΙΜΟΛΟΓΙΟ**  
**Αρ.Παραστατικού : 2**  
**Ημερομ.Έκδοσης : 2018-04-10**

**ΠΕΛΑΤΗΣ**  
**ΕΠΙΧΕΙΡΗΣΗ ΑΕ**  
**ΕΜΠΟΡΙΟ**  
**ΑΦΜ: 146113015 ΔΟΥ: ΝΕΑΣ ΙΩΝΙΑΣ**  
**ΑΡΤΑΣ 1**  
**ΑΘΗΝΑ Τ.Κ: 14231**

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ
APPLE IPHONE	1.0	800.0	13.0	800.0
SAMSUNG TABLET	3.0	256.0	24.0	768.0
ΦΠΑ 24				184.32
ΦΠΑ 13				104.0
ΣΥΝΟΛΟ ΦΠΑ				288.32
ΣΥΝΟΛΟ ΠΡΟΙΟΝΤΩΝ				1568.0
ΤΕΛΙΚΟ ΣΥΝΟΛΟ				1856.32

Εικόνα 4.36 Το τιμολόγιο

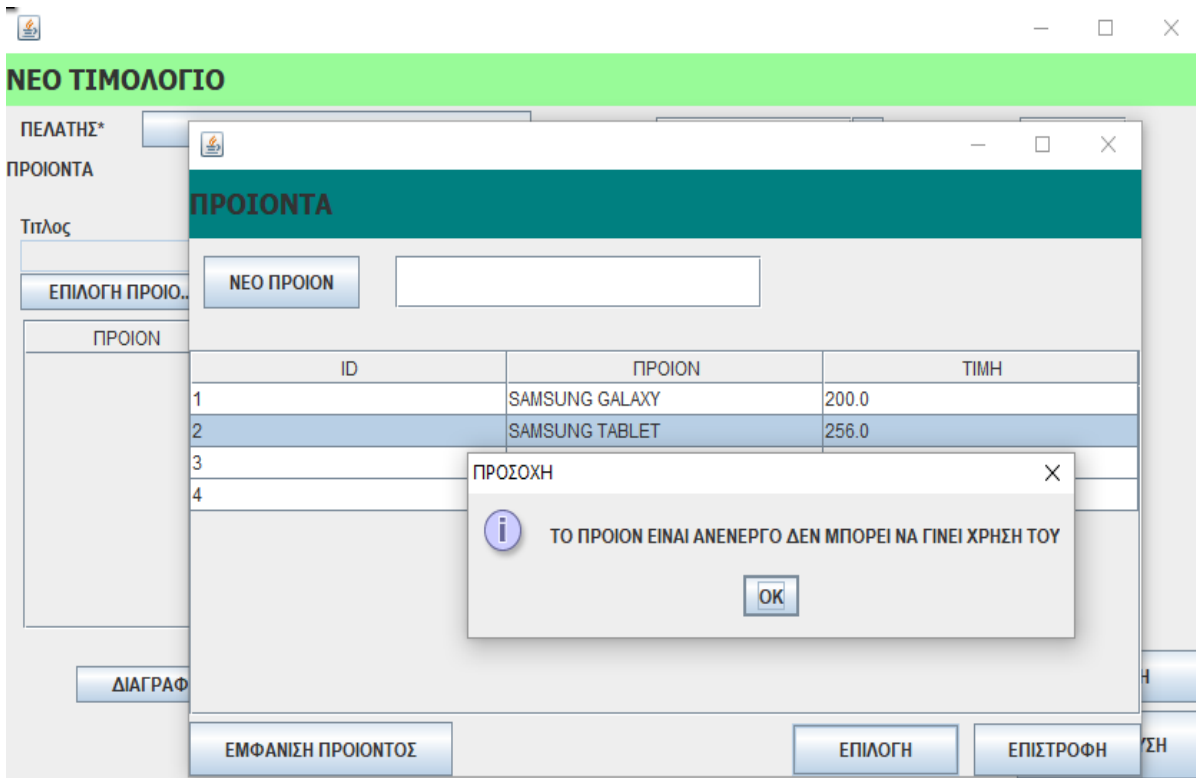
#### 4.5.3 Νέο τιμολόγιο

Στην οθόνη νέο τιμολόγιο με το κουμπί επιλογή πελάτη εμφανίζεται οθόνη για αναζήτηση του πελατολογίου. Σε περίπτωση που δεν έχει επιλεγεί πελάτης εμφανίζεται μήνυμα που ενημερώνει το χρήστη. Στο πεδίο ημερομηνία έχουμε αυτόματη συμπλήρωση της σημερινής ημερομηνίας και στο πεδίο αριθμός παραστατικού τον αριθμό που θα πάρει το τιμολόγιο.

Εικόνα 4.37 Οθόνη νέο τιμολόγιο

Με το κουμπί επιλογή προϊόντος εμφανίζεται η οθόνη προϊόντα για την αναζήτηση του επιθυμητού προϊόντος. Σε περίπτωση που ένα προϊόν είναι ανενεργό και γίνει επιλογή του εμφανίζεται μήνυμα ενημερώνοντας το χρήστη ότι δεν μπορεί να γίνει χρήση του.





Εικόνα 4.38 Ανενεργό προϊόν

Κατά την επιλογή προϊόντος μπορεί να γίνει αλλαγή της ποσότητας, της τιμής και του Φ.Π.Α. Για την καταχώρηση του πατάμε το κουμπί προσθήκη. Σε περίπτωση που θέλουμε να διαγράψουμε ένα προϊόν που έχουμε καταχωρήσει πατάμε το κουμπί διαγραφή. (εικόνες 4.39 & 4.40)

Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

**ΝΕΟ ΤΙΜΟΛΟΓΙΟ**

ΠΕΛΑΤΗΣ\* ΠΑΤΣΗΣ ΜΑΡΙΟΣ ΗΜΕΡ\* 10/04/2018 Α.ΠΑΡΑΣΤ\* 4

ΠΡΟΙΟΝΤΑ

Τιτλος Ποσοτητα Τιμη ΦΠΑ

APPLE IPHONE 1 800 13 ΠΡΟΣΘΗΚΗ

ΕΠΙΛΟΓΗ ΠΡΟΙΟ...

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ

ΔΙΑΓΡΑΦΗ

ΦΠΑ 24 : ΣΥΝ. ΠΛΗΝ ΦΠΑ :  
 ΦΠΑ 13 : ΤΕΛΙΚΟ ΣΥΝΟΛΟ :  
 ΦΠΑ 6 :

ΑΚΥΡΩΣΗ  
 ΑΠΟΘΗΚΕΥΣΗ

Εικόνα 4.39 Επιλογή προϊόντος

**ΝΕΟ ΤΙΜΟΛΟΓΙΟ**

ΠΕΛΑΤΗΣ\* ΠΑΤΣΗΣ ΜΑΡΙΟΣ ΗΜΕΡ\* 10/04/2018 Α.ΠΑΡΑΣΤ\* 4

ΠΡΟΙΟΝΤΑ

Τιτλος Ποσοτητα Τιμη ΦΠΑ

  1 1 13 ΠΡΟΣΘΗΚΗ

ΕΠΙΛΟΓΗ ΠΡΟΙΟ...

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ
APPLE IPHONE	1.0	800.0	13	800.0

ΔΙΑΓΡΑΦΗ

ΦΠΑ 24 : ΣΥΝ. ΠΛΗΝ ΦΠΑ : 800.0  
 ΦΠΑ 13 : 104.0 ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 904.0  
 ΦΠΑ 6 :

ΑΚΥΡΩΣΗ  
 ΑΠΟΘΗΚΕΥΣΗ

Εικόνα 4.40 Καταχώρηση επιλεγμένου προϊόντος

Επιλογή αλλού προϊόντος το οποίο έχει ενεργοποιημένη την αποθήκη. Η εφαρμογή δείχνει το διαθέσιμο απόθεμα.(εικόνα 4.41)

**ΝΕΟ ΤΙΜΟΛΟΓΙΟ**

ΠΕΛΑΤΗΣ\* ΠΑΤΣΙΗΣ ΜΑΡΙΟΣ ΗΜΕΡ\* 10/04/2018 Α.ΠΑΡΑΣΤ\* 4

ΠΡΟΙΟΝΤΑ

Τίτλος Ποσότητα Τιμή ΦΠΑ

APPLE IPAD 4 3 349 24 **ΠΡΟΣΘΗΚΗ**

**ΕΠΙΛΟΓΗ ΠΡΟΙΟ...** Διαθεσ.Αποθ: 7.0

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ
APPLE IPHONE	1.0	800.0	13	800.0

ΔΙΑΓΡΑΦΗ

ΦΠΑ 24 : ΣΥΝ. ΠΛΗΝ ΦΠΑ : 800.0  
ΦΠΑ 13 : 104.0 ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 904.0  
ΦΠΑ 6 :

**ΑΚΥΡΩΣΗ**  
**ΑΠΟΘΗΚΕΥΣΗ**

Εικόνα 4.41 Προϊόν με απόθεμα

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

**ΝΕΟ ΤΙΜΟΛΟΓΙΟ**

ΠΕΛΑΤΗΣ\* ΠΑΤΣΗΣ ΜΑΡΙΟΣ ΗΜΕΡ\* 10/04/2018 Α.ΠΑΡΑΣΤ\* 4

ΠΡΟΙΟΝΤΑ

Τιτλος  Ποσότητα 1 Τιμη 1 ΦΠΑ 24

Δαθεσ.Αποθ :

ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ	ΦΠΑ	ΣΥΝΟΛΟ
APPLE IPHONE	1.0	800.0	13	800.0
APPLE IPAD 4	3.0	349.0	24	1047.0

ΦΠΑ 24 : 251.28 ΣΥΝ. ΠΛΗΝ ΦΠΑ : 1847.0  
 ΦΠΑ 13 : 104.0 ΤΕΛΙΚΟ ΣΥΝΟΛΟ : 2202.28  
 ΦΠΑ 6 :

Εικόνα 4.42 Καταχώρηση δευτέρου προϊόντος

**ΤΙΜΟΛΟΓΙΑ**

ID	ΕΠΩΝΥΜΙΑ	ΑΦΜ	ΗΜ.ΕΚΔΟΣΗΣ	ΠΟΣΟ	ΠΛΗΡΩΜΕΝΟ
4	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015	2018-04-10	2202.28	ΟΧΙ
3	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	146113015	2018-04-09	0.0	ΟΧΙ
2	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	146113015	2018-04-10	1856.32	ΟΧΙ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	146113015	2018-04-10	496.0	ΟΧΙ

Εικόνα 4.43 Αποθήκευση τιμολογίου

## 4.6 Αναφορές

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

Στην οθόνη αναφορές μπορούμε να έχουμε μια συνοπτική εικόνα για τους πελάτες, τις πωλήσεις και τα προϊόντα



Εικόνα 4.44 Οθόνη αναφορές

#### 4.6.1 Αναφορές επαφής

Για να γίνει μια αναφορά επιλέγουμε το επιθυμητό τύπο αναφοράς, το εύρος ημερομηνίας και το κουμπί αναζήτηση για να εκτελεστεί. (εικόνα 4.45)

ΕΠΩΝΥΜΙΑ	ΑΝΕΞΟΦΛΗΤΑ ΤΙΜΟΛΟΓΙΑ	ΣΥΝ.ΠΟΣΟ
----------	----------------------	----------

Εικόνα 4.45 Αναφορές επαφών

#### Υπόλοιπο επαφών

Στο τύπο αναφοράς υπόλοιπο επαφών, μπορούμε να δούμε στοιχεία σχετικά με τα ανεξόφλητα παραστατικά ανά πελάτη. (εικόνα 4.46)

The screenshot shows a software window titled "ΑΝΑΦΟΡΕΣ ΕΠΑΦΩΝ". On the left, there is a vertical menu with buttons for "ΕΠΑΦΗΣ", "ΠΩΛΗΣΕΩΝ", "ΠΡΟΙΟΝΤΩΝ", and "ΕΠΙΣΤΡΟΦΗ". The main area contains a search form with the following fields: "Τυπος Αναφορας" (set to "ΥΠΟΛΟΙΠΟ ΕΠΑΦΩΝ"), "Απο" (10/04/2018), and "Εως" (10/04/2018). Below the search form is a table with the following data:

ID ΠΕΛΑΤΗ	ΕΠΩΝΥΜΙΑ	ΑΝΕΞΟΦΛΗΤΑ ΤΙΜΟΛΟΓΙΑ	ΣΥΝ.ΠΟΣΟ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	2	2698.28
2	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	1	1856.32

At the bottom of the window, there is a summary row showing "3" and "4554".

Εικόνα 4.46 Υπόλοιπο επαφών

### Μεγαλύτερο ποσό

Στο τύπο αναφοράς μεγαλύτερο ποσό, μπορούμε να δούμε στοιχεία για πελάτες ταξινομώντας τους με βάση τις περισσότερες πωλήσεις που έχουν γίνει. (εικόνα 4.47)

The screenshot shows the same software window "ΑΝΑΦΟΡΕΣ ΕΠΑΦΩΝ". The search form is now set to "ΜΕΓΑΛΥΤΕΡΟ ΠΟΣΟ", with "Απο" (10/04/2018) and "Εως" (24/04/2018). The table below shows the same data as in the previous screenshot:

ID ΠΕΛΑΤΗ	ΕΠΩΝΥΜΙΑ	ΑΡ.ΤΙΜΟΛΟΓΙΩΝ	ΣΥΝ.ΠΟΣΟ
1	ΠΑΤΣΗΣ ΜΑΡΙΟΣ	2	2698.28
2	ΕΠΙΧΕΙΡΗΣΗ ΑΕ	1	1856.32

Εικόνα 4.47 Μεγαλύτερο ποσό

#### 4.6.2 Αναφορές πωλήσεων

Για να γίνει μια αναφορά επιλέγουμε το επιθυμητό τύπο αναφοράς, το εύρος ημερομηνίας που και το κουμπί αναζήτηση για να εκτελεστεί. (εικόνα 4.48)

ID TIM	ΦΠΑ 6	ΦΠΑ 13	ΦΠΑ 24	ΚΑΘΑΡΗ ΑΞΙΑ	ΤΕΛΙΚΗ ΑΞΙΑ
--------	-------	--------	--------	-------------	-------------

Εικόνα 4.48 Αναφορές πωλήσεων

#### Παραστατικά πωλήσεων

Στον τύπο αναφοράς παραστατικά πωλήσεων, μπορούμε να δούμε στοιχεία σχετικά με τα παραστατικά που έχουν εκδοθεί. (εικόνα 4.49)



ΑΝΑΦΟΡΕΣ ΠΩΛΗΣΕΩΝ						
ΕΠΑΦΗΣ	Τυπος Αναφοράς	Απο	Εως			
	ΠΑΡΑΣΤΑΤΙΚΑ ΠΩΛΗΣΕΩΝ	10/04/2018	10/04/2018	ΑΝΑΖΗΤ...		
ΠΩΛΗΣΕΩΝ	ID TIM	ΦΠΑ 6	ΦΠΑ 13	ΦΠΑ 24	ΚΑΘΑΡΗ ΑΞΙΑ	ΤΕΛΙΚΗ ΑΞΙΑ
	1	0.0	0.0	96.0	400.0	496.0
	2	0.0	104.0	184.32	1568.0	1856.32
	4	0.0	104.0	251.28	1847.0	2202.28
ΠΡΟΙΟΝΤΩΝ	ΑΡ.ΤΙΜ: 3	ΣΥΝ. :0.0	ΣΥΝ. :208.0	ΣΥΝ. :531.6	ΣΥΝ. :3815.0	ΣΥΝ. :4554.6
ΕΠΙΣΤΡΟΦΗ						

Εικόνα 4.49 Παραστατικά πωλήσεων

#### 4.6.3 Αναφορές προϊόντων

Για να γίνει μια αναφορά επιλέγουμε το επιθυμητό τύπο αναφοράς, το εύρος ημερομηνίας που και το κουμπί αναζήτηση για να εκτελεστεί. (εικόνα 4.50)

ID ΠΡΟΙΟΝΤΟΣ	ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ ΠΩΛΗΣΕΩΝ	ΜΕΣΗ ΤΙΜΗ	ΣΥΝΟΛΟ ΠΩΛΗΣΕΩΝ
--------------	--------	-------------------	-----------	-----------------

Εικόνα 4.50 Αναφορές προϊόντων

### Πωλήσεις προϊόντων

Στον τύπο αναφοράς πωλήσεις προϊόντων , μπορούμε να δούμε στοιχεία σχετικά με τα προϊόντα που έχουν πουληθεί. (εικόνα 4.51)

## Ανάπτυξη εφαρμογής έκδοσης παραστατικών για εμπορική επιχείρηση

ID ΠΡΟΙΟΝΤΟΣ	ΠΡΟΙΟΝ	ΠΟΣΟΤΗΤΑ ΠΩΛΗ...	ΜΕΣΗ ΤΙΜΗ	ΚΑΘΑΡΗ ΑΞΙΑ	ΣΥΝΟΛΟ ΤΕΛΙΚΟ
1	SAMSUNG GALAXY	2.0	200.0	400.0	400.0
4	APPLE IPHONE	2.0	800.0	1600.0	1808.0
2	SAMSUNG TABLET	3.0	256.0	768.0	768.0
3	APPLE IPAD 4	3.0	349.0	1047.0	1047.0

Εικόνα 4.51 Πωλήσεις προϊόντων



## ΚΕΦΑΛΑΙΟ 5

### ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ

#### 5.1 Δημιουργία βάσης δεδομένων

```
CREATE DATABASE IF NOT EXISTS `invoice`
USE `invoice`;

-- table invoice.contacts
CREATE TABLE IF NOT EXISTS `contacts` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `CompanyName` text,
  `Occupation` text,
  `VatNumber` text,
  `TaxOffice` text,
  `StreetAddress` text,
  `Town` text,
  `Zip` text,
  `Email` text,
  `Tel` text,
  `Mobile` text,
  `Status` text,
  `Dcreated` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `Dmodified` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- table invoice.email
CREATE TABLE IF NOT EXISTS `email` (
  `ID` int(11) DEFAULT NULL,
  `Email` text,
  `Pass` text,
  `Smtpserver` text,
  `Smtpport` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- table invoice.inventory
CREATE TABLE IF NOT EXISTS `inventory` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `ID_Inventory` int(11) DEFAULT NULL,
  `Total` double unsigned DEFAULT NULL,
  `DateEdited` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`ID`),
  UNIQUE KEY `ID_Inventory` (`ID_Inventory`),
  CONSTRAINT `FK_inventory_products` FOREIGN KEY (`ID_Inventory`)
REFERENCES `products` (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

```
-- table invoice.invoiceitems
CREATE TABLE IF NOT EXISTS `invoiceitems` (
  `InvoiceID` int(11) DEFAULT NULL,
  `ProductID` int(11) NOT NULL,
  `ProductName` text,
  `Quantity` double DEFAULT NULL,
  `Price` double DEFAULT NULL,
  `Tax` double DEFAULT NULL,
  `DateInserted` date DEFAULT NULL,
  KEY `FK_invoiceitems_invoices` (`InvoiceID`),
  KEY `FK_invoiceitems_products` (`ProductID`),
  CONSTRAINT `FK_invoiceitems_invoices` FOREIGN KEY (`InvoiceID`)
REFERENCES `invoices` (`ID`),
  CONSTRAINT `FK_invoiceitems_products` FOREIGN KEY (`ProductID`)
REFERENCES `products` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- table invoice.invoicepdf
CREATE TABLE IF NOT EXISTS `invoicepdf` (
  `InvoiceID` int(11) DEFAULT NULL,
  `Pdf` mediumblob,
  KEY `FK_invoicepdf_invoices` (`InvoiceID`),
```

```
CONSTRAINT `FK_invoicepdf_invoices` FOREIGN KEY (`InvoiceID`) REFERENCES  
`invoices` (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--table invoice.invoices  
CREATE TABLE IF NOT EXISTS `invoices` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `ClientID` int(11) DEFAULT NULL,  
  `InvoiceDate` date DEFAULT NULL,  
  `Status` varchar(50) DEFAULT NULL,  
  `Total` double unsigned DEFAULT NULL,  
  `PaidStatus` enum('Y','N') DEFAULT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `FK_invoices_contacts` (`ClientID`),  
  CONSTRAINT `FK_invoices_contacts` FOREIGN KEY (`ClientID`) REFERENCES  
`contacts` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

```
-- table invoice.mycompany  
CREATE TABLE IF NOT EXISTS `mycompany` (  
  `ID` int(1) NOT NULL AUTO_INCREMENT,  
  `Name` text,  
  `CTitle` text,  
  `Occupation` text,  
  `Vat` text,  
  `Doy` text,  
  `Street` text,  
  `Town` text,  
  `ZipCode` text,  
  `tel1` text,  
  `tel2` text,  
  `fax` text,  
  `mobile` text,  
  `email` text,  
  `logo` longblob,
```

```
PRIMARY KEY (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
  
-- table invoice.products  
CREATE TABLE IF NOT EXISTS `products` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Name` text,  
  `ProdCode` text,  
  `Description` text,  
  `Price` double unsigned DEFAULT '0',  
  `Tax` int(2) unsigned DEFAULT NULL,  
  `Sales` enum('T','F') DEFAULT NULL,  
  `Stock` enum('T','F') DEFAULT NULL,  
  `DateCreated` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `DateEdited` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

## 5.2 Πακετο Util

### 5.2.1 Main.java

```
package util;  
import java.awt.BorderLayout;  
import java.awt.EventQueue;  
import javax.swing.*;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.border.EmptyBorder;  
  
import products.ProductsG;  
import reports.ReportsG;  
import settings.SettingsG;  
import contacts.Epafes;  
import invoice.InvoiceG;  
  
import javax.swing.JLabel;  
import javax.swing.JButton;  
import java.awt.event.ActionListener;  
import java.io.IOException;  
import java.sql.SQLException;  
import java.awt.event.ActionEvent;
```



```
import java.awt.Font;
import java.awt.Color;

public class Main {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Main m=new Main();

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

    /**
     * Create the frame.
     */
    public Main() {
        JFrame frame= new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setBounds(100, 100, 537, 355);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        frame.setContentPane(contentPane);
        contentPane.setLayout(null);
        frame.setVisible(true);
        JLabel lblNewLabel = new JLabel("ΑΡΧΙΚΗ");
        lblNewLabel.setBackground(new Color(222, 184, 135));
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBounds(0, 0, 521, 45);
        contentPane.add(lblNewLabel);

        JButton btnepafes = new JButton("ΕΠΑΦΕΣ");
        btnepafes.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Epafes epafes= new Epafes();
            }
        });
    }
}
```

```

        epafes.setVisible(true);
    }
});
btnepafes.setBounds(0, 56, 132, 71);
contentPane.add(btnepafes);

JButton button = new JButton("ΠΡΟΙΟΝΤΑ");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ProductsG productsg=new ProductsG();
        productsg.setVisible(true);
    }
});
button.setBounds(0, 137, 132, 71);
contentPane.add(button);

JButton btnsettings = new JButton("ΡΥΘΜΙΣΕΙΣ");
btnsettings.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        SettingsG settingsg = null;
        try {
            settingsg = new SettingsG();
        } catch (SQLException |
IOException e1) {
            // TODO Auto-generated catch
block
            e1.printStackTrace();
        }
        settingsg.setVisible(true);
    }
});
btnsettings.setBounds(0, 219, 132, 71);
contentPane.add(btnsettings);

JButton btninvoice = new JButton("ΤΙΜΟΛΟΓΙΑ");
btninvoice.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        InvoiceG invoiceg = new InvoiceG();
        invoiceg.setVisible(true);
    }
});
btninvoice.setBounds(142, 56, 132, 71);
contentPane.add(btninvoice);

JButton button_1 = new JButton("ΑΝΑΦΟΡΕΣ");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```
        ReportsG reportsg=new ReportsG();
        reportsg.setVisible(true);
    }
});
button_1.setBounds(142, 137, 132, 71);
contentPane.add(button_1);
}
}
```

### 5.2.2 MySqlConn.java

```
package util;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.*.*;

import javax.swing.JOptionPane;

public class MySqlConn {

    public static Connection dbConnector() throws SQLException {
        Connection conn = null;
        try {
            String driver = "com.mysql.jdbc.Driver";
            Class.forName(driver);
            String url =
"jdbc:mysql://localhost:3306/invoice?useUnicode=true&characterEncoding=UTF-8";
            conn = DriverManager.getConnection(url,"root","1234");
            return conn;
        }catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}
```

## 5.3 Πακέτο contacts

### 5.3.1 Epafes.java

```
package contacts;
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;
import javax.swing.JToggleButton;
import javax.swing.JMenuBar;
import javax.swing.JList;
import javax.swing.border.MatteBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import java.awt.Color;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumnModel;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.ListSelectionModel;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.AbstractAction;
import javax.swing.Action;
import java.awt.Font;

public class Epafes extends JFrame {

    public JPanel contentPane;
    private JTextField textField;
    public JTable table;
    public JButton btnselect;

    public Epafes() {

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 681, 419);
        contentPane = new JPanel();
```

```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblNewLabel = new JLabel("ΕΠΑΦΕΣ");
lblNewLabel.setOpaque(true);
lblNewLabel.setBackground(Color.LIGHT_GRAY);
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel.setForeground(Color.BLACK);
lblNewLabel.setBounds(0, 0, 665, 43);
contentPane.add(lblNewLabel);

JButton btnnewcontact = new JButton("ΝΕΑ ΕΠΑΦΗ");
btnnewcontact.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        NewContact newcontact = new NewContact();
        newcontact.setVisible(true);
    }
});
btnnewcontact.setBounds(10, 54, 109, 33);
contentPane.add(btnnewcontact);

textField = new JTextField();
textField.addKeyListener(new KeyAdapter() {
    @Override
    public void keyTyped(KeyEvent e) {
        if(textField.getText().length() >= 0)
        {
            findUsers();
        }else {
            table.setModel(new DefaultTableModel(
                new Object[][] {
                },
                new String[] {
                    "ID", "ΕΠΩΝΥΜΙΑ", "ΑΦΜ"
                }
            ));
        }
    }
});

textField.setBounds(151, 54, 256, 33);
contentPane.add(textField);
textField.setColumns(10);
```

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(0, 113, 665, 229);
contentPane.add(scrollPane);

table = new JTable();
table.setRowHeight(20);

table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
table.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "ID", "ΕΠΩΝΥΜΙΑ", "ΑΦΜ"
    }
));

TableModel tcm = table.getColumnModel();
tcm.getColumn(0).setPreferredWidth(150);
tcm.getColumn(1).setPreferredWidth(265);
tcm.getColumn(2).setPreferredWidth(250);

scrollPane.setViewportView(table);

JButton btnNewButton = new JButton("ΕΜΦΑΝΙΣΗ ΕΠΑΦΗΣ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        opencontact();

    }
});

btnNewButton.setBounds(0, 346, 163, 34);
contentPane.add(btnNewButton);
btnNewButton.setEnabled(false);

JButton btnNewButton_1 = new JButton("ΕΠΙΣΤΡΟΦΗ");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        dispose();
    }
});
btnNewButton_1.setBounds(548, 347, 117, 33);
contentPane.add(btnNewButton_1);
```

```
        btnselect = new JButton("ΕΠΙΛΟΓΗ");
        btnselect.setVisible(false);
        btnselect.setBounds(429, 347, 109, 33);
        contentPane.add(btnselect);

        table.getSelectionModel().addListSelectionListener(new
ListSelectionListener(){
            public void valueChanged(ListSelectionEvent e) {
                //enable button - put it in an EDT to be safe
though
                ListSelectionModel lsm =
(ListSelectionModel)e.getSource();
                if (lsm.isSelectionEmpty()) {
                    btnNewButton.setEnabled(false);
                }else {
                    btnNewButton.setEnabled(true);
                }
            }
        });

    }

    public void findUsers()
    {
        ArrayList<ContactsD> users =
ContactsD.ListUsers(textField.getText());
        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(new
Object[]{"ID", "ΕΠΩΝΥΜΙΑ", "ΑΦΜ"});

        Object[] row = new Object[3];

        for(int i = 0; i < users.size(); i++)
        {
            row[0] = users.get(i).getId();
            row[1] = users.get(i).getCname();
            row[2] = users.get(i).getAfm();
            model.addRow(row);
        }

        table.setModel(model);
    }
}
```

```
}

    public void opencontact()
    {
        int column = 0;
        int row = table.getSelectedRow();
        String value = table.getModel().getValueAt(row,
column).toString();

        System.out.print(value);
        ArrayList<ContactsD> contact =
ContactsD.FindContact(value);

        ContactPG contactpg = new ContactPG();
        contactpg.setVisible(true);
        contactpg.PreviewContact(contact);

    }
}
```

### 5.3.2 NewContact.java

```
package contacts;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Window;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JSeparator;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JFormattedTextField;
import javax.swing.JOptionPane;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.KeyAdapter;
import javax.swing.event.AncestorListener;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.xml.ws.Holder;

import gr.gspr.webservicealldatadesktopclient.ServiceClient;
```



```
import
gr.gspr.webservicealldatadesktopclient.client.RgWsPublicBasicRtUse
rs;

import javax.swing.event.AncestorEvent;
import javax.swing.ImageIcon;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.ArrayList;
import java.awt.Font;
import java.awt.Color;

public class NewContact extends JFrame {

    public JPanel contentPane;
    public JTextField tcn;
    public JTextField toc;
    static JTextField tafm;
    public JTextField tdoy;
    public JTextField tstreet;
    public JTextField ttown;
    public JTextField ttk;
    public JTextField temail;
    public JTextField ttel;
    public JTextField tmob;
    public JLabel lblNewLabel_1;
    public int id=0;

    public ServiceClient sc = new ServiceClient();
    public String un="MMMMPPPP93";
    public String pass="MMMMPPPP93";
    public String o=" ";
    public JLabel lblNewLabel;

    public NewContact() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 883, 472);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel_1 = new JLabel("");
        lblNewLabel_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
```

```
tcn.setText(sc.basicRtUser.value.getOnomasia());

tstreet.setText(sc.basicRtUser.value.getPostalAddress().trim() + sc.basicRtUser.value.getPostalAddressNo().trim() );

ttk.setText(sc.basicRtUser.value.getPostalZipCode().trim());

ttown.setText(sc.basicRtUser.value.getPostalAreaDescription().trim());

tdoy.setText(sc.basicRtUser.value.getDoyDescr());
    }
});

lblNewLabel_1.setIcon(new
ImageIcon(NewContact.class.getResource("/icons/if_check_1930264.png")));
lblNewLabel_1.setBounds(161, 233, 28, 32);
contentPane.add(lblNewLabel_1);
lblNewLabel_1.setVisible(false);

JLabel label_15 = new JLabel("");
label_15.setBounds(161, 233, 28, 32);
contentPane.add(label_15);
label_15.setVisible(false);

tafm = new JTextField();
tafm.getDocument().addDocumentListener(new
DocumentListener() {

    @Override
    public void insertUpdate(DocumentEvent e) {
        // TODO Auto-generated method stub

        if(tafm.getText().length()==9) {
            try {
                if(checktAfm()) {
                    lblNewLabel_1.setVisible(true);
                }else {

lblNewLabel_1.setVisible(false);
                }

            } catch (Exception e1) {
```

```
block // TODO Auto-generated catch
      e1.printStackTrace();
    }
  }else {
    lblNewLabel_1.setVisible(false);
  }
}
@Override
public void removeUpdate(DocumentEvent e) {
  // TODO Auto-generated method stub
  if(tafm.getText().length()==9) {
    try {
      if(checktAfm()) {
        lblNewLabel_1.setVisible(true);
      }else {
        lblNewLabel_1.setVisible(false);
      }
    } catch (Exception e1) {
      // TODO Auto-generated catch
      e1.printStackTrace();
    }
  }else {
    lblNewLabel_1.setVisible(false);
  }
}
}
@Override
public void changedUpdate(DocumentEvent e) {
  // TODO Auto-generated method stub
}
});

  tafm.setColumns(10);
  tafm.setBounds(10, 233, 137, 32);
```

```
contentPane.add(tafm);

lblNewLabel = new JLabel("ΔΗΜΙΟΥΡΓΙΑ ΕΠΑΦΗΣ");
lblNewLabel.setBackground(Color.LIGHT_GRAY);
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel.setOpaque(true);
lblNewLabel.setBounds(0, 0, 867, 54);
contentPane.add(lblNewLabel);

tcn = new JTextField();
tcn.setBounds(10, 85, 374, 32);
contentPane.add(tcn);
tcn.setColumns(10);

JLabel label = new JLabel("ΕΠΩΝΥΜΙΑ ΕΠΙΧΕΙΡΗΣΗΣ");
label.setBounds(10, 65, 133, 14);
contentPane.add(label);

JLabel label_1 = new JLabel("ΔΡΑΣΤΗΡΙΟΤΗΤΑ");
label_1.setBounds(10, 139, 133, 14);
contentPane.add(label_1);

toc = new JTextField();
toc.setColumns(10);
toc.setBounds(10, 160, 374, 32);
contentPane.add(toc);

JLabel label_2 = new JLabel("ΑΦΜ");
label_2.setBounds(10, 214, 90, 14);
contentPane.add(label_2);

JLabel label_3 = new JLabel("ΔΟΥ");
label_3.setBounds(202, 214, 133, 14);
contentPane.add(label_3);

tdoy = new JTextField();
tdoy.setColumns(10);
tdoy.setBounds(195, 233, 189, 32);
contentPane.add(tdoy);

JLabel label_4 = new JLabel("ΔΙΕΥΘΥΝΣΗ");
label_4.setBounds(10, 288, 133, 14);
contentPane.add(label_4);

tstreet = new JTextField();
tstreet.setColumns(10);
tstreet.setBounds(10, 312, 236, 32);
contentPane.add(tstreet);
```

```
JLabel label_5 = new JLabel("ΠΟΛΗ");
label_5.setBounds(261, 288, 133, 14);
contentPane.add(label_5);

ttown = new JTextField();
ttown.setColumns(10);
ttown.setBounds(261, 312, 236, 32);
contentPane.add(ttown);

JLabel label_6 = new JLabel("ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΑΣ");
label_6.setBounds(10, 360, 133, 14);
contentPane.add(label_6);

ttk = new JTextField();
ttk.setColumns(10);
ttk.setBounds(10, 385, 102, 32);
contentPane.add(ttk);

JSeparator separator = new JSeparator();
separator.setBounds(537, 112, 0, -58);
contentPane.add(separator);

JLabel lblEmail = new JLabel("EMAIL");
lblEmail.setBounds(553, 65, 46, 14);
contentPane.add(lblEmail);

temail = new JTextField();
temail.setColumns(10);
temail.setBounds(552, 85, 306, 32);
contentPane.add(temail);

JLabel label_7 = new JLabel("ΤΗΛΕΦΩΝΟ");
label_7.setBounds(553, 139, 122, 14);
contentPane.add(label_7);

ttel = new JTextField();
ttel.setColumns(10);
ttel.setBounds(552, 160, 306, 32);
contentPane.add(ttel);

JLabel label_8 = new JLabel("ΚΙΝΗΤΟ");
label_8.setBounds(553, 214, 46, 14);
contentPane.add(label_8);

tmob = new JTextField();
tmob.setColumns(10);
tmob.setBounds(552, 239, 306, 32);
contentPane.add(tmob);
```

```
        JButton btnNewButton = new JButton("ΕΦΑΡΜΟΓΗ");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (checkInput()){
                    if(id==0) {

                        ContactsD.addEntry(tcn,toc,tafm,tdoy,tstreet,ttown,ttk,temail,ttel,tmob);

                                dispose();
                            }
                        else {
                            ContactsD.updateEntry(id, tcn,
                                toc, tafm, tdoy, tstreet, ttown, ttk, temail, ttel, tmob);
                                dispose();
                            }
                        }
                    else {

                        JOptionPane.showMessageDialog(null,
                            "ΕΙΣΑΓΕΤΕ ΟΛΑ ΤΑ ΑΠΑΙΤΟΥΜΕΝΑ ΚΕΛΙΑ", "ΠΡΟΣΟΧΗ",
                            JOptionPane.INFORMATION_MESSAGE);
                    }
                }
            });
        btnNewButton.setBounds(745, 385, 102, 36);
        contentPane.add(btnNewButton);

        JButton button = new JButton("ΑΚΥΡΩΣΗ");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                dispose();
            }
        });
        button.setBounds(633, 385, 102, 36);
        contentPane.add(button);

        JLabel label_9 = new JLabel("");
        label_9.setBounds(388, 85, 46, 14);
        contentPane.add(label_9);

        JLabel label_10 = new JLabel("");
        label_10.setBounds(388, 160, 46, 14);
        contentPane.add(label_10);

        JLabel label_11 = new JLabel("");
        label_11.setBounds(151, 233, 46, 14);
        contentPane.add(label_11);
```

```
JLabel label_12 = new JLabel("");
label_12.setBounds(388, 233, 46, 14);
contentPane.add(label_12);

JLabel label_13 = new JLabel("");
label_13.setBounds(246, 312, 46, 14);
contentPane.add(label_13);

JLabel label_14 = new JLabel("");
label_14.setBounds(505, 312, 46, 14);
contentPane.add(label_14);

}

public boolean checkInput() {

    if (tcn.getText().isEmpty()) return false;
    if (toc.getText().isEmpty()) return false;
    if (tafm.getText().isEmpty()) return false;
    if (tdoy.getText().isEmpty()) return false;
    if (tstreet.getText().isEmpty()) return false;
    if (ttown.getText().isEmpty()) return false;

    return true;
}

public boolean checkAfm() throws Exception {

    String rr = sc.getData(un, pass, tafm.getText(),o);
    //System.out.println(rr);

    if(sc.basicRtUser.value.getAfm()!=null) {
        return true;
    }else {
        return false;
    }
}

public void setForModify(ArrayList<ContactsD> contact) {

    for(int i = 0; i < contact.size(); i++)
    {

        tcn.setText(contact.get(i).getCname());
    }
}
```

```
        toc.setText(contact.get(i).getOc());
        tafm.setText(contact.get(i).getAfm());
        tdoy.setText(contact.get(i).getDoy());
        tstreet.setText(contact.get(i).getStreet());
        ttown.setText(contact.get(i).getTown());
        ttk.setText(contact.get(i).getZip());
        temail.setText(contact.get(i).getEmail());
        ttel.setText(contact.get(i).getTel());
        tmob.setText(contact.get(i).getMob());
        //lblstatus1.setText(contact.get(i).getStatus());

        id=contact.get(i).getId();
    }

}

}
```

### 5.3.3 ContactPG.java

```
package contacts;
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.util.ArrayList;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;

import invoice.InvoiceD;

import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.ListSelectionModel;

public class ContactPG extends JFrame {

    private JPanel contentPane;
    private final JLabel lblNewLabel = new JLabel("ΕΠΑΦΗ");
```



```
public JLabel lblcn1;
public JLabel lbloc1;
public JLabel lblafm1;
public JLabel lbldoy1;
public JLabel lblstreet1;
public JLabel lbltown1;
public JLabel lbltk1;
public JLabel lblemail1;
public JLabel lbltel1;
public JLabel lblmob1;
public JLabel lblstatus1;
public JLabel lbldcreated1;
public JLabel lbldmodified1;
public JLabel lblid1;

public static int id;
public ArrayList<ContactsD> contact;
private JButton btnNewButton;
private JScrollPane scrollPane;
private JTable table;

public ContactPG() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 681, 417);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    lblNewLabel.setBackground(Color.LIGHT_GRAY);
    lblNewLabel.setOpaque(true);
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel.setBounds(0, 0, 665, 47);
    contentPane.add(lblNewLabel);

    JLabel lblcn = new JLabel("ΕΠΩΝΥΜΙΑ :");
    lblcn.setBounds(10, 59, 71, 14);
    contentPane.add(lblcn);

    JLabel lbloc = new JLabel("ΔΡΑΣΤΗΡΙΟΤΗΤΑ :");
    lbloc.setBounds(10, 81, 107, 14);
    contentPane.add(lbloc);

    JLabel lblafm = new JLabel("ΑΦΜ:");
    lblafm.setBounds(10, 106, 71, 14);
    contentPane.add(lblafm);

    JLabel lbldoy = new JLabel("ΔΟΥY :");
```

```
lblDoy.setBounds(10, 131, 71, 14);
contentPane.add(lblDoy);

JLabel lblStreet = new JLabel("ΔΙΕΥΘΥΝΣΗ :");
lblStreet.setBounds(10, 156, 71, 14);
contentPane.add(lblStreet);

JLabel lblTown = new JLabel("ΠΟΛΗ :");
lblTown.setBounds(10, 181, 71, 14);
contentPane.add(lblTown);

JLabel lblTk = new JLabel("ΤΑΧ.Κ :");
lblTk.setBounds(10, 206, 71, 14);
contentPane.add(lblTk);

JLabel lblEmail = new JLabel("email :");
lblEmail.setBounds(10, 231, 71, 14);
contentPane.add(lblEmail);

JLabel lblTel = new JLabel("ΤΗΛ :");
lblTel.setBounds(10, 256, 71, 14);
contentPane.add(lblTel);

JLabel lblMob = new JLabel("ΚΙΝ :");
lblMob.setBounds(10, 280, 71, 14);
contentPane.add(lblMob);

lblcn1 = new JLabel("");
lblcn1.setBounds(91, 59, 218, 14);
contentPane.add(lblcn1);

lbloc1 = new JLabel("");
lbloc1.setBounds(111, 81, 218, 14);
contentPane.add(lbloc1);

lblafm1 = new JLabel("");
lblafm1.setBounds(91, 106, 218, 14);
contentPane.add(lblafm1);

lblDoy1 = new JLabel("");
lblDoy1.setBounds(91, 131, 218, 14);
contentPane.add(lblDoy1);

lblStreet1 = new JLabel("");
lblStreet1.setBounds(91, 156, 218, 14);
contentPane.add(lblStreet1);

lblTown1 = new JLabel("");
lblTown1.setBounds(91, 181, 218, 14);
```

```
contentPane.add(lbltown1);

    lbltk1 = new JLabel("");
    lbltk1.setBounds(91, 206, 218, 14);
    contentPane.add(lbltk1);

    lblemail1 = new JLabel("");
    lblemail1.setBounds(91, 231, 218, 14);
    contentPane.add(lblemail1);

    lbltel1 = new JLabel("");
    lbltel1.setBounds(91, 256, 218, 14);
    contentPane.add(lbltel1);

    lblmob1 = new JLabel("");
    lblmob1.setBounds(91, 280, 218, 14);
    contentPane.add(lblmob1);

    lblcreated1 = new JLabel("");
    lblcreated1.setBounds(242, 318, 218, 14);
    contentPane.add(lblcreated1);

    lblmodified1 = new JLabel("");
    lblmodified1.setBounds(10, 318, 218, 14);
    contentPane.add(lblmodified1);

    JButton btnedit = new JButton("ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙΩΝ");
    btnedit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            NewContact newcontact = new NewContact();
            newcontact.setVisible(true);
            newcontact.setForModify(contact);
            newcontact.lblNewLabel.setText("ΕΠΕΞΕΡΓΑΣΙΑ
ΕΠΑΦΗΣ");
        }
    });
    btnedit.setBounds(0, 343, 188, 35);
    contentPane.add(btnedit);

    btnNewButton = new JButton("ΕΠΙΣΤΡΟΦΗ");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            dispose();
        }
    });
```

```
        btnNewButton.setBounds(543, 343, 122, 35);
        contentPane.add(btnNewButton);

        JLabel lblNewLabel_1 = new JLabel("ΜΗ ΕΞΟΦΛΗΜΕΝΑ
ΤΙΜΟΛΟΓΙΑ");
        lblNewLabel_1.setBounds(382, 59, 198, 14);
        contentPane.add(lblNewLabel_1);

        scrollPane = new JScrollPane();
        scrollPane.setBounds(382, 74, 283, 260);
        contentPane.add(scrollPane);

        table = new JTable();

        table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        table.setModel(new DefaultTableModel(
            new Object[][] {
            },
            new String[] {
                "ID", "ΗΜ.ΕΚΔΟΣΗΣ", "ΠΟΣΟ"
            }
        ));
        scrollPane.setViewportViewView(table);
        scrollPane.setEnabled(false);

        JLabel label = new JLabel("Ημ.Τροποποίησης");
        label.setBounds(14, 305, 142, 14);
        contentPane.add(label);

        JLabel label_1 = new JLabel("Ημ. Δημιουργίας");
        label_1.setBounds(242, 305, 142, 14);
        contentPane.add(label_1);

    }

    public void PreviewContact(ArrayList<ContactsD> contact) {
        this.contact=contact;

        for(int i = 0; i < contact.size(); i++)
        {
            //lblid1.setText(contact.get(i).getId());
            lblcn1.setText(contact.get(i).getCname());
            lbloc1.setText(contact.get(i).getOc());
            lblafm1.setText(contact.get(i).getAfm());
            lbldoy1.setText(contact.get(i).getDoy());
            lblstreet1.setText(contact.get(i).getStreet());
            lbltown1.setText(contact.get(i).getTown());
        }
    }
}
```

```
        lbltk1.setText(contact.get(i).getZip());
        lblemail1.setText(contact.get(i).getEmail());
        lbltel1.setText(contact.get(i).getTel());
        lblmob1.setText(contact.get(i).getMob());
        //lblstatus1.setText(contact.get(i).getStatus());

        lblcreated1.setText(contact.get(i).getDcreated());

        lblmodified1.setText(contact.get(i).getDmodified());

        id=contact.get(i).getId();

    }

    filltable();
}

public void filltable() {

    ArrayList<InvoiceD> invoices =
InvoiceD.selectUnpaidInvoices(id);

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new
Object[]{"ID", "ΗΜ.ΕΚΔΟΣΗΣ", "ΠΟΣΟ"});

    for(int i = 0; i < invoices.size(); i++)
    {
        Object[] row = new Object[3];

        row[0] = invoices.get(i).getId();
        row[1] = invoices.get(i).getInvoiceDate();
        row[2] = invoices.get(i).getTotal();
        model.addRow(row);
    }
    table.setModel(model);

}
}
```

### 5.3.4 ContactsD.java

```
package contacts;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Blob;
import java.sql.ResultSet;
import java.sql.SQLException;

import java.sql.Statement;
import java.util.ArrayList;

import javax.swing.*;

import com.mysql.jdbc.PreparedStatement;

import util.MySqlConnection;

public class ContactsD extends NewContact{

    private int id;
    private String cname;
    private String afm;
    private String oc;
    private String doy;
    private String street;
    private String mob;
    private String town;
    private String email;
    private String tel;
    private String zip;
    private String dcreated;
    private String dmodified;
    private String status;

    private String ctitle;
    private String tel2;
    private byte[] logo;
    private String fax;
    public byte[] buffer = new byte[1024];
    private String spor;
    private String sser;
    private String pas;
    private String em;
```

```
public ContactsD(int Id,String CName,String Afm)
{
    this.id = Id;
    this.cname = CName;
    this.afm = Afm;
}

public ContactsD(int Id,String CName,String Oc,String
Afm,String Doy,String Street,String Town,String Zip,String
Email,String Tel,String Mob,String Status,String Dcreated,String
Dmodified)
{
    this.id = Id;
    this.cname = CName;
    this.oc = Oc;
    this.afm = Afm;
    this.doy = Doy;
    this.street = Street;
    this.town = Town;
    this.zip = Zip;
    this.email = Email;
    this.tel = Tel;
    this.mob = Mob;
    this.status=Status;
    this.dcreated = Dcreated;
    this.dmodified = Dmodified;
}

public ContactsD() {
    // TODO Auto-generated constructor stub
}

public int getId()
{
    return id;
}

public String getCname()
{
    return cname;
}

public String getOc()
{
    return oc;
}
```

```
public String getAfm()
{
    return afm;
}

public String getDoy()
{
    return doy;
}

public String getStreet()
{
    return street;
}
public String getTown()
{
    return town;
}

public String getZip()
{
    return zip;
}
public String getEmail()
{
    return email;
}
public String getTel()
{
    return tel;
}
public String getMob()
{
    return mob;
}
public String getStatus()
{
    return status;
}
public String getDcreated()
{
    return dcreated;
}

public String getDmodified()
{
    return dmodified;
}
```



```
public String getTel2()
{
    return tel2;
}

public String getCtitle()
{
    return ctitle;
}

public byte[] getLogo()
{
    return logo;
}

public String getFax()
{
    return fax;
}

public String getEm()
{
    return em;
}

public String getPas()
{
    return pas;
}

public String getSser()
{
    return sser;
}

public String getSpor()
{
    return spor;
}
```

```
public static void addEntry(JTextField tcn, JTextField toc,
JTextField tafm, JTextField tdoy, JTextField tstreet, JTextField
```

```
ttown, JTextField ttk, JTextField temail, JTextField ttel,
JTextField tmob) {

    try {
        Statement stmt =
MySqlConnection.dbConnector().createStatement();

        String sql = "INSERT INTO
contacts(ID,CompanyName,Occupation,VatNumber,TaxOffice,StreetAddre
ss,Town,Zip,Email,Tel,Mobile,Status) VALUES ("
            + "NULL" + ", '" +
            tcn.getText() + "', '" +
            toc.getText() + "', '" +
            tafm.getText() + "', '" +
            tdoy.getText() + "', '" +
            tstreet.getText() + "', '" +
            ttown.getText() + "', '" +
            ttk.getText() + "', '" +
            temail.getText() + "', '" +
            ttel.getText() + "', '" +
            tmob.getText() + "', 'ACTIVE')";

            stmt.executeUpdate(sql);

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}
```

```
public static void updateEntry(int id, JTextField tcn, JTextField
toc, JTextField tafm, JTextField tdoy, JTextField tstreet,
JTextField ttown, JTextField ttk, JTextField temail, JTextField
ttel, JTextField tmob) {
```

```
    try {
        Statement stmt =
MySqlConnection.dbConnector().createStatement();

        String sql = "UPDATE contacts SET
CompanyName=' "
```

```
        + tcn.getText() + "', Occupation='"
        + toc.getText() + "', VatNumber='"
        + tafm.getText() + "', TaxOffice='"
+ tdoy.getText() + "', StreetAddress='"
+ tstreet.getText() + "', Town='"
+ ttown.getText() + "', Zip='"
+ ttk.getText() + "', Email='"
+ temail.getText() + "', Tel='"
+ ttel.getText() + "', Mobile='"
+ tmob.getText() + "' WHERE ID="+ id ;

        stmt.executeUpdate(sql);

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

public static ArrayList<ContactsD> ListUsers(String ValToSearch)
{
    ArrayList<ContactsD> usersList = new ArrayList<ContactsD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `contacts` WHERE
(`CompanyName` LIKE '%" + ValToSearch + "%' OR `VatNumber` LIKE
 '%" + ValToSearch + "%' )";
        rs = stmt.executeQuery(searchQuery);

        ContactsD user;

        while(rs.next())
        {
            user = new ContactsD(
                rs.getInt("ID"),
                rs.getString("CompanyName"),
                rs.getString("VatNumber")
            );
            usersList.add(user);
        }
    }
```

```
    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }

    return userList;
}

public static ArrayList<ContactsD> FindContact(String ValToSearch)
{
    ArrayList<ContactsD> userList = new ArrayList<ContactsD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `contacts` WHERE
(`ID` = '"+ValToSearch+"' )";
        rs = stmt.executeQuery(searchQuery);

        ContactsD user;

        while(rs.next())
        {
            user = new ContactsD(
                rs.getInt("ID"),
                rs.getString("CompanyName"),
                rs.getString("Occupation"),
                rs.getString("VatNumber"),
                rs.getString("TaxOffice"),
                rs.getString("StreetAddress"),
                rs.getString("Town"),
                rs.getString("Zip"),
                rs.getString("Email"),
                rs.getString("Tel"),
                rs.getString("Mobile"),
                rs.getString("Status"),
                rs.getString("Dcreated"),
                rs.getString("Dmodified")
            );

            userList.add(user);
        }

    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }
}
```

```
    return userList;
}
```

```
public static void updateMyCompany(JTextField tcn, JTextField tct,
JTextField toc, JTextField tafm, JTextField tdoy, JTextField
tstreet, JTextField ttown, JTextField ttk, JTextField temail,
JTextField ttel, JTextField ttel2, JTextField tmob, JTextField
tfax, File selectedFile) throws FileNotFoundException {
```

```
    FileInputStream fis=new FileInputStream(selectedFile);

    try {
        Statement stmt =
MySqlConnection.dbConnector().createStatement();

        PreparedStatement statement = null;

        String sql = "UPDATE mycompany SET Name='"
            + tcn.getText() + "', CTitle='"
            + tct.getText() + "', Occupation='"
            + toc.getText() + "', Vat='"
            + tafm.getText() + "', Doy='"
            + tdoy.getText() + "', Street='"
            + tstreet.getText() + "', Town='"
            + ttown.getText() + "', ZipCode='"
            + ttk.getText() + "', Email='"
            + temail.getText() + "', Tel1='"
            + ttel.getText() + "', Tel2='"
            + ttel2.getText() + "', Mobile='"
            + tmob.getText() + "', fax='"
            + tfax.getText() + "' WHERE ID=1" ;

            stmt.executeUpdate(sql);
            statement = (PreparedStatement)
MySqlConnection.dbConnector().prepareStatement("UPDATE mycompany SET
logo= ? WHERE ID=1");

            statement.setBinaryStream(1,(InputStream)fis,(int)(selectedF
ile.length()));

            statement.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
```

```
        e.printStackTrace();
    }

}

public void getMyCompany() {

    ResultSet rs;

    try{
        Statement stmt =
        MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `mycompany`
WHERE `ID`=1";
        rs = stmt.executeQuery(searchQuery);

        //Blob blob;

        while(rs.next())
        {
            this.id=rs.getInt("ID");

this.cname=rs.getString("Name");

this.ctitle=rs.getString("CTitle");

this.oc=rs.getString("Occupation");
            this.afm=rs.getString("Vat");
            this.doy=rs.getString("Doy");

this.street=rs.getString("Street");
            this.town=
rs.getString("Town");

this.zip=rs.getString("ZipCode");
            this.tel=rs.getString("tel1");
            this.tel2=rs.getString("tel2");
            this.fax=rs.getString("fax");

this.mob=rs.getString("mobile");

this.email=rs.getString("email");
            this.logo=rs.getBytes("logo");

System.out.println(this.logo);

```

```
    }  
  
    }catch(Exception ex){  
        System.out.println(ex.getMessage());  
    }  
  
}  
  
public void updateEmail(JTextField textField_13, JTextField  
textField_14, JTextField textField_15, JTextField textField_16) {  
  
    try {  
        Statement stmt =  
MySqlConnection.dbConnector().createStatement();  
  
        String sql = "UPDATE email SET Email='"  
            + textField_13.getText() + "', Pass='"  
            + textField_14.getText() + "', Smtserver='"  
            + textField_15.getText() + "', Smtpport='"  
            + textField_16.getText() + "' WHERE ID=1" ;  
  
        stmt.executeUpdate(sql);  
  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
  
}  
  
public void getSEmail() {  
  
    ResultSet rs;  
  
    try{  
        Statement stmt =  
MySqlConnection.dbConnector().createStatement();  
        String searchQuery = "SELECT * FROM `email` WHERE  
`ID`=1";  
        rs = stmt.executeQuery(searchQuery);
```

```
        while(rs.next())
        {
                                this.em=rs.getString("Email");
                                this.pas=rs.getString("Pass");

this.sser=rs.getString("Smtserver");

this.spor=rs.getString("Smtport");

        }

        }catch(Exception ex){
            System.out.println(ex.getMessage());
        }

}

}
```

## 5.4 Πακέτο products

### 5.4.1 ProductsG.java

```
package products;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.ArrayList;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.ListSelectionModel;
import javax.swing.border.EmptyBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.DefaultTableModel;
import java.awt.Font;
```



```
import java.awt.Color;

public class ProductsG extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    public JTable table;
    public JButton btnok;
    public int l;

    public ProductsG() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 681, 419);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("ΠΡΟΙΟΝΤΑ");
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBackground(new Color(0, 128, 128));
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setBounds(0, 0, 665, 43);
        contentPane.add(lblNewLabel);

        JButton btnnewproduct = new JButton("NEO ΠΡΟΙΟΝ");
        btnnewproduct.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                NewProductG newproductg = new NewProductG();
                newproductg.setVisible(true);
            }
        });
        btnnewproduct.setBounds(10, 54, 109, 33);
        contentPane.add(btnnewproduct);

        textField = new JTextField();
        textField.addKeyListener(new KeyAdapter() {
            @Override
            public void keyTyped(KeyEvent e) {
                if(textField.getText().length() >= 0)
                {
                    findProducts();
                }else {
                    table.setModel(new DefaultTableModel(
new Object[][] {
},
new String[] {
"ID", "ΟΝΟΜΑ ΠΡΟΙΟΝΤΟΣ", "ΠΕΡΙΓΡΑΦΗ"

```

```
        }
    ));
        }
    });

/*   textField.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            findUsers();

        }
    }); */

textField.setBounds(144, 54, 256, 33);
contentPane.add(textField);
textField.setColumns(10);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(0, 113, 665, 229);
contentPane.add(scrollPane);

table = new JTable();
table.setRowHeight(20);

table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
table.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "ID", "ΠΡΟΙΟΝ", "ΤΙΜΗ"
    }
));
scrollPane.setViewportViewView(table);

JButton btnNewButton = new JButton("ΕΜΦΑΝΙΣΗ
ΠΡΟΙΟΝΤΟΣ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        openproduct();

    }
});

btnNewButton.setBounds(0, 346, 184, 34);
contentPane.add(btnNewButton);
```

```
        btnNewButton.setEnabled(false);

        JButton btnNewButton_1 = new JButton("ΕΠΙΣΤΡΟΦΗ");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                dispose();
            }
        });
        btnNewButton_1.setBounds(548, 347, 117, 33);
        contentPane.add(btnNewButton_1);

        btnok = new JButton("ΕΠΙΛΟΓΗ");
        btnok.setBounds(421, 347, 117, 33);
        btnok.setVisible(false);
        contentPane.add(btnok);

        table.getSelectionModel().addListSelectionListener(new
        ListSelectionListener(){
            public void valueChanged(ListSelectionEvent e) {
                //enable button - put it in an EDT to be safe
            }
        });
    }

    public void findProducts()
    {
        ArrayList<ProductsD> products =
        ProductsD.ListProducts(textField.getText());
        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(new
        Object[]{"ID", "ΠΡΟΙΟΝ", "ΤΙΜΗ"});
        Object[] row = new Object[3];

        for(int i = 0; i < products.size(); i++)
        {
```

```
        row[0] = products.get(i).getId();
        row[1] = products.get(i).getName();
        row[2] = products.get(i).getPrice();

        if(products.get(i).getSales()==false)

            model.addRow(row);
    }
    table.setModel(model);
}

public void openproduct()
{
    int column = 0;
    int row = table.getSelectedRow();
    String value = table.getModel().getValueAt(row,
column).toString();

    System.out.print(value);
    ArrayList<ProductsD> product =
ProductsD.FindProductsD(value);

    ProductPG productpg = new ProductPG();
    productpg.setVisible(true);
    productpg.PreviewProduct(product);

}
}
```

#### 5.4.2 NewProductG.java

```
package products;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
```

```
import javax.swing.DefaultComboBoxModel;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.awt.event.ActionEvent;
import java.awt.Color;
import java.awt.Font;

public class NewProductG extends JFrame {

    private JPanel contentPane;
    public JLabel lblNewLabel = new JLabel("NEO ΠΡΟΙΟΝ");
    private JTextField tname;
    private JTextField tpcode;
    private JTextField tprice;
    private JTextField tdes;
    private JComboBox comboBox;
    private JCheckBox tstock;
    public int id=0;

    public NewProductG() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 815, 472);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setBackground(new Color(0, 128, 128));
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBounds(0, 0, 799, 45);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("ΟΝΟΜΑ ΠΡΟΙΟΝΤΟΣ");
        lblNewLabel_1.setBounds(0, 45, 175, 14);
        contentPane.add(lblNewLabel_1);

        tname = new JTextField();
        tname.setBounds(0, 65, 308, 27);
        contentPane.add(tname);
        tname.setColumns(10);

        JLabel label = new JLabel("ΚΩΔΙΚΟΣ ΠΡΟΙΟΝΤΟΣ");
        label.setBounds(0, 109, 175, 14);
        contentPane.add(label);

        tpcode = new JTextField();
        tpcode.setColumns(10);
```

```
tpcode.setBounds(0, 129, 308, 27);
contentPane.add(tpcode);

JLabel label_1 = new JLabel("ΠΕΡΙΓΡΑΦΗ ΠΡΟΙΟΝΤΟΣ");
label_1.setBounds(0, 174, 175, 14);
contentPane.add(label_1);

tdes = new JTextField();
tdes.setColumns(10);
tdes.setBounds(0, 194, 308, 27);
contentPane.add(tdes);

tprice = new JTextField();
tprice.setColumns(10);
tprice.setBounds(0, 262, 120, 27);
contentPane.add(tprice);

JLabel label_2 = new JLabel("ΤΙΜΗ");
label_2.setBounds(0, 242, 175, 14);
contentPane.add(label_2);

JLabel label_3 = new JLabel("Φ.Π.Α");
label_3.setBounds(130, 242, 175, 14);
contentPane.add(label_3);

comboBox = new JComboBox();
comboBox.setModel(new DefaultComboBoxModel(new String[]
{"24", "13", "6"}));
comboBox.setBounds(130, 262, 142, 27);
contentPane.add(comboBox);

tstock = new JCheckBox("ΑΠΟΘΗΚΗ");
tstock.setBounds(6, 312, 114, 23);
contentPane.add(tstock);

JButton btnNewButton = new JButton("ΕΦΑΡΜΟΓΗ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(checkInput()) {
            if(id==0) {

ProductsD.addEntry(tname,tpcode,tdes,tprice,comboBox,tstock)
;
                dispose();
            }else
            {
                ProductsD.updateEntry(id,
tname,tpcode,tdes,tprice,comboBox,tstock);
            }
        }
    }
});
```

```
        dispose();
    }
    }else
    {
        JOptionPane.showMessageDialog(null,
"ΕΙΣΑΓΕΤΕ ΟΛΑ ΤΑ ΑΠΑΙΤΟΥΜΕΝΑ ΚΕΛΙΑ", "ΠΡΟΣΟΧΗ",
JOptionPane.INFORMATION_MESSAGE);
    }
}
});
btnNewButton.setBounds(669, 396, 130, 37);
contentPane.add(btnNewButton);

JButton button = new JButton("ΑΚΥΡΩΣΗ");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
button.setBounds(526, 396, 130, 37);
contentPane.add(button);
}
```

```
public boolean checkInput() {
```

```
    if (tname.getText().isEmpty()) return false;
    //if (toc.getText().isEmpty()) return false;
```

```
    return true;
```

```
}
```

```
public void setForModify(ArrayList<ProductsD> product) {
```

```
    for(int i = 0; i < product.size(); i++)
    {
```

```
        tname.setText(product.get(i).getName());
        tpcode.setText(product.get(i).getProdCode());
        tdes.setText(product.get(i).getDescription());
```

```
        tprice.setText(String.valueOf(product.get(i).getPrice()));
```

```
        if(product.get(i).getTax()==24) {
```

```
        comboBox.setSelectedIndex(0);
    }
    if(product.get(i).getTax()==13){
        comboBox.setSelectedIndex(1);
    }
    if(product.get(i).getTax()==6){
        comboBox.setSelectedIndex(2);
    }

    if(product.get(i).getStock() {
        tstock.setSelected(true);
    }else {
        tstock.setSelected(false);
    }

    id=product.get(i).getId();
}

}

}
```

### 5.4.3 ProductPG.java

```
package products;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.EventQueue;
import java.awt.Window;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JToggleButton;
import javax.swing.JSeparator;
import javax.swing.AbstractButton;
import javax.swing.JButton;
```



```
import java.awt.event.ActionListener;
import java.text.ParseException;
import java.util.ArrayList;
import java.awt.event.ActionEvent;
import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;
import java.awt.Font;
import java.awt.Color;

public class ProductPG extends JFrame {

    private JPanel contentPane;
    public JLabel label8;
    public JLabel label9;
    private JLabel label10;
    private JLabel label5;
    private JLabel label7;
    private JToggleButton toggle;
    private JLabel label13;
    private JLabel label14;

    public static int id;
    public ArrayList<ProductsD> product;
    private AbstractButton toggleButton;
    private JLabel lblNewLabel_1;
    private JLabel stockav;
    private JLabel label;
    private JLabel lblled;
    private JSpinner spinner;
    private JButton btnNewButton_1;
    private JButton btnNewButton_2;

    public ProductPG() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 681, 417);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("ΠΡΟΙΟΝ");
        lblNewLabel.setBackground(new Color(0, 128, 128));
        lblNewLabel.setOpaque(true);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setBounds(0, 0, 665, 40);
        contentPane.add(lblNewLabel);

        JLabel label1 = new JLabel("ΟΝΟΜΑ ΠΡΟΪΟΝΤΟΣ :");
        label1.setBounds(10, 61, 130, 14);
```

```
contentPane.add(label1);

JLabel label2 = new JLabel("ΚΩΔΙΚΟΣ ΠΡΟΪΟΝΤΟΣ :");
label2.setBounds(10, 85, 130, 14);
contentPane.add(label2);

JLabel label3 = new JLabel("ΠΕΡΙΓΡΑΦΗ ΠΡΟΪΟΝΤΟΣ :");
label3.setBounds(10, 110, 140, 14);
contentPane.add(label3);

JLabel label4 = new JLabel("ΤΙΜΗ :");
label4.setBounds(10, 145, 58, 14);
contentPane.add(label4);

label5 = new JLabel("");
label5.setBounds(78, 145, 97, 14);
contentPane.add(label5);

JLabel label6 = new JLabel("Φ.Π.Α :");
label6.setBounds(10, 172, 58, 14);
contentPane.add(label6);

label7 = new JLabel("");
label7.setBounds(78, 172, 97, 14);
contentPane.add(label7);

toggle = new JToggleButton("ΕΝΕΡΓΟ");
toggle.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if(toggle.isSelected()) {
            toggle.setSelected(true);
            toggle.setText("ΕΝΕΡΓΟ");

            ProductsD.updateSS('T',"Sales",id);

        }else {
            toggle.setSelected(false);
            toggle.setText("ΑΝΕΝΕΡΓΟ");
            ProductsD.updateSS('F',"Sales",id);

        }

    }
});
toggle.setSelected(true);
toggle.setBounds(0, 197, 156, 23);
contentPane.add(toggle);
```

```
label8 = new JLabel("");
label8.setBounds(150, 61, 185, 14);
contentPane.add(label8);

label9 = new JLabel("");
label9.setBounds(150, 85, 185, 14);
contentPane.add(label9);

label10 = new JLabel("");
label10.setBounds(150, 110, 185, 14);
contentPane.add(label10);

JButton button = new JButton("ΔΙΟΡΘΩΣΗ ΣΤΟΙΧΕΙΩΝ");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        NewProductG newproduct = new NewProductG();
        newproduct.setVisible(true);
        newproduct.setForModify(product);
    }
});
button.setBounds(0, 343, 156, 35);
contentPane.add(button);

JButton button_1 = new JButton("ΕΠΙΣΤΡΟΦΗ");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
button_1.setBounds(543, 343, 122, 35);
contentPane.add(button_1);

JLabel label11 = new JLabel("ΗΜ. ΔΗΜΙΟΥΡΓΙΑΣ : ");
label11.setBounds(10, 274, 130, 14);
contentPane.add(label11);

JLabel label12 = new JLabel("ΗΜ. ΤΕΛΕΥΤ ΤΡΟΠΟΠ :");
label12.setBounds(10, 299, 130, 14);
contentPane.add(label12);

label13 = new JLabel("");
label13.setBounds(150, 274, 185, 14);
contentPane.add(label13);

label14 = new JLabel("");
label14.setBounds(150, 299, 185, 14);
contentPane.add(label14);
```

```
toggleButton = new JToggleButton("ΑΠΟΘΗΚΗ");
toggleButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("mbbbbbbb");
        if(toggleButton.isSelected()) {
            toggleButton.setSelected(true);
            toggleButton.setText("ΑΠΟΘΗΚΗ
ΕΝΕΡΓΗ");

                ProductsD.updateSS('T',"Stock",id);

                ProductsD.showInventory(id);

            stockav.setText((String.valueOf(ProductsD.total)));

            lblled.setText(ProductsD.dateeditedinventory);
                stockChangeG(0);

        }else {
            toggleButton.setSelected(false);
            toggleButton.setText("ΑΠΟΘΗΚΗ
ΑΝΕΝΕΡΓΗ");

                ProductsD.updateSS('F',"Stock",id);

            toggleButton.setSelected(false);
            toggleButton.setText("ΑΠΟΘΗΚΗ
ΑΝΕΝΕΡΓΗ");

                stockChangeG(1);
        }

    }
});
toggleButton.setSelected(true);
toggleButton.setBounds(0, 231, 156, 23);
contentPane.add(toggleButton);

lblNewLabel_1 = new JLabel("ΔΙΑΘΕΣΙΜΗ ΑΠΟΘΗΚΗ :");
lblNewLabel_1.setBounds(388, 61, 153, 14);
contentPane.add(lblNewLabel_1);

stockav = new JLabel("");
stockav.setBounds(551, 61, 104, 14);
contentPane.add(stockav);

spinner = new JSpinner();
```

```
((JSpinner) spinner).setModel(new
SpinnerNumberModel(new Double(0), new Double(0), null, new
Double(1)));
((Component) spinner).setBounds(482, 107, 76, 20);
contentPane.add((Component) spinner);

btnNewButton_1 = new JButton("ΠΡΟΣΘΗΚΗ");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        double value = (Double) spinner.getValue();

ProductsD.updateInventory(value+ProductsD.total,id);
        ProductsD.showInventory(id);

stockav.setText((String.valueOf(ProductsD.total)));

lblEd.setText(ProductsD.dateeditedinventory);

    }
});
btnNewButton_1.setBounds(411, 141, 89, 23);
contentPane.add(btnNewButton_1);

btnNewButton_2 = new JButton("ΑΦΑΙΡΕΣΗ");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        double value = (Double) spinner.getValue();

        if(value<=ProductsD.total) {
ProductsD.updateInventory(ProductsD.total-
value,id);
            ProductsD.showInventory(id);

stockav.setText((String.valueOf(ProductsD.total)));

lblEd.setText(ProductsD.dateeditedinventory);
        }else {
            JOptionPane.showMessageDialog(null,
"ΛΑΘΟΣ ΣΤΗΝ ΑΦΑΙΡΕΣΗ ΠΡΟΙΟΝΤΩΝ ΑΠΟ ΑΠΟΘΗΚΗ", "ΠΡΟΣΟΧΗ",
JOptionPane.INFORMATION_MESSAGE);
        }

    }
});
btnNewButton_2.setBounds(510, 141, 89, 23);
contentPane.add(btnNewButton_2);
```

```
        label = new JLabel("ΗΜ. ΤΕΛΕΥΤ ΑΛΛΑΓΗΣ  :");
        label.setBounds(388, 172, 130, 14);
        contentPane.add(label);

        lblled = new JLabel("");
        lblled.setBounds(442, 206, 198, 14);
        contentPane.add(lblled);
    }

    public void PreviewProduct(ArrayList<ProductsD> product) {
        this.product=product;

        for(int i = 0; i < product.size(); i++)
        {
            id=product.get(i).getId();

            label8.setText(product.get(i).getName());
            label9.setText(product.get(i).getProdCode());
            label10.setText(product.get(i).getDescription());

            label15.setText(String.valueOf(product.get(i).getPrice()));

            label17.setText(String.valueOf(product.get(i).getTax()));

            if(product.get(i).getSales() {
                toggle.setSelected(true);
                toggle.setText("ΕΝΕΡΓΟ");
            }else{
                toggle.setSelected(false);
                toggle.setText("ΑΝΕΝΕΡΓΟ");
            }

            if(product.get(i).getStock() {
                toggleButton.setSelected(true);
                toggleButton.setText("ΑΠΟΘΗΚΗ ΕΝΕΡΓΗ");
                ProductsD.showInventory(id);

            stockav.setText((String.valueOf(ProductsD.total)));

            lblled.setText(ProductsD.dateeditedinventory);
                stockChangeG(0);
            }else{
                toggleButton.setSelected(false);
                toggleButton.setText("ΑΠΟΘΗΚΗ ΑΝΕΝΕΡΓΗ");
            }
        }
    }
}
```

```
        stockChangeG(1);
    }

    label13.setText(product.get(i).getDatecreated());
    label14.setText(product.get(i).getDateEdited());

}

}

private void stockChangeG(int a) {

    if (a==0) { //true

        lblNewLabel_1.setEnabled(true);
        stockav.setEnabled(true);
        spinner.setEnabled(true);
        btnNewButton_1.setEnabled(true);
        btnNewButton_2.setEnabled(true);
        label.setEnabled(true);
        lblEd.setEnabled(true);

    }else {
        lblNewLabel_1.setEnabled(false);
        stockav.setEnabled(false);
        spinner.setEnabled(false);
        btnNewButton_1.setEnabled(false);
        btnNewButton_2.setEnabled(false);
        label.setEnabled(false);
        lblEd.setEnabled(false);
    }

}

}
```

#### 5.4.4 ProductsD.java

```
package products;

import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.sql.Statement;
import java.util.ArrayList;

import javax.swing.*;

import util.MySqlConnection;

public class ProductsD {

    private int id;
    private String name;
    private double price;
    private String prodcode;
    private String description;
    private int tax;
    private boolean sales;
    private boolean stock;
    private String datecreated;
    private String dateedited;
    public static double total;
    public static String dateeditedinventory;

    public ProductsD(int Id,String Name,double Price)
    {
        this.id = Id;
        this.name = Name;
        this.price = Price;
    }

    public ProductsD(int Id,String Name,String ProdCode,String
Description,double Price,int Tax,boolean Sales,boolean
Stock,String DateCreated,String DateEdited)
    {
        this.id = Id;
        this.name = Name;
        this.prodcode =ProdCode ;
        this.description = Description;
        this.price = Price;
        this.tax = Tax;
        this.sales = Sales;
        this.stock = Stock;
    }
}
```



```
        this.datecreated = DateCreated;
        this.dateedited = DateEdited;

    }

    public int getId()
    {
        return id;
    }

    public String getName()
    {
        return name;
    }

    public String getProdCode()
    {
        return prodcode;
    }

    public String getDescription()
    {
        return description;
    }

    public double getPrice()
    {
        return price;
    }

    public int getTax()
    {
        return tax;
    }

    public boolean getSales()
    {
        return sales;
    }
    public boolean getStock()
    {
        return stock;
    }

    public String getDatecreated()
```

```
{
    return datecreated;
}

public String getDateEdited()
{
    return dateedited;
}

public double getTotal() {
    return total;
}

public String getDateEditedInventory()
{
    return dateeditedinventory;
}

public static void addEntry(JTextField tname, JTextField tpcode,
JTextField tdes, JTextField tprice, JComboBox comboBox, JCheckBox
tstock) {

    char s;

    String cmb = (String) comboBox.getSelectedItemAt();
    if (tstock.isSelected()){
        s = 'T';
    }else {
        s='F';
    }

    try {
        Statement stmt =
MySQLConn.dbConnector().createStatement();

        String sql = "INSERT INTO
products(ID,Name,ProdCode,Description,Price,Tax,Sales,Stock)
VALUES ("
                + "NULL" +",','"+
                tname.getText() + "',' ' +
                tpcode.getText() + "',' ' +
                tdes.getText() + "',' ' +
                tprice.getText() + "',' ' +
                cmb + "',' ' +
                "T" + "',' ' +
                s+ "',')";
```

```
        stmt.executeUpdate(sql
,Statement.RETURN_GENERATED_KEYS);

        ResultSet rs =
stmt.getGeneratedKeys();
        if(rs.next())
        {
            int last_inserted_id =
rs.getInt(1);
            addInventory(last_inserted_id,0);
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
public static void updateEntry(int id ,JTextField tname,
JTextField tpcode, JTextField tdes, JTextField tprice, JComboBox
comboBox, JCheckBox tstock) {
```

```
    char s;
    String cmb = (String) comboBox.getSelectedItem();
    if (tstock.isSelected()){
        s = 'T';
    }else {
        s='F';
    }

    try {
        Statement stmt = MySqlConnection.dbConnector().createStatement();

        String sql = "UPDATE products SET Name='"
            + tname.getText() + "', ProdCode='"
            + tpcode.getText() + "', Description='"
            + tdes.getText() + "', Price='"
            + tprice.getText() + "', Tax='"
            + cmb + "', Stock='"
            + s + "' WHERE ID="+ id ;

            stmt.executeUpdate(sql);
```

```
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

public static void updateSS(char t,String r,int id) {

    try {
        Statement stmt =
        MySqlConnection.dbConnector().createStatement();
        String sql = "UPDATE products SET "+ r +"='\" + t +\"' WHERE
        ID="+ id ;

            stmt.executeUpdate(sql);

    }

        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

public static void addInventory(int id,double z) {

    try {
        Statement stmt =
        MySqlConnection.dbConnector().createStatement();

        String sql = "INSERT INTO
        inventory(ID,ID_Inventory,Total) VALUES ("
            + "NULL" +", '\"+
            id + "\", '\" +
            z +\"')\";

            stmt.executeUpdate(sql);
    }
}
```

```
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

public static void updateInventory(double t,int id) {

    System.out.println(t);
    try {
        Statement stmt =
        MySqlConnection.dbConnector().createStatement();
        String sql = "UPDATE inventory SET Total='" + t +"' WHERE
        ID_Inventory="+ id ;
            stmt.executeUpdate(sql);
            total=t;

    }

        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

public static double showInventory(int id) {
    ResultSet rs;

    try {
        Statement stmt =
        MySqlConnection.dbConnector().createStatement();

        String searchQuery = "SELECT * FROM `inventory` WHERE
        (`ID_Inventory`= '"+id+"' )";
        rs = stmt.executeQuery(searchQuery);

        while(rs.next())
        {

            total= rs.getDouble("Total");
            dateeditedinventory =
rs.getString("DateEdited");
```

```
    }  
} catch (SQLException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
  
return total;  
}
```

```
public static ArrayList<ProductsD> ListProducts(String  
ValToSearch)  
{  
    ArrayList<ProductsD> productsList = new  
ArrayList<ProductsD>();  
  
    ResultSet rs;  
  
    try{  
        Statement stmt = MySqlConnection.dbConnector().createStatement();  
        String searchQuery = "SELECT * FROM `products` WHERE  
(`Name` LIKE '%" + ValToSearch + "%' OR `ProdCode` LIKE  
'" + ValToSearch + "%' )";  
        rs = stmt.executeQuery(searchQuery);  
  
        ProductsD product;  
  
        while(rs.next())  
        {  
            product = new ProductsD(  
                rs.getInt("ID"),  
                rs.getString("Name"),  
                rs.getDouble("Price")  
            );  
            productsList.add(product);  
        }  
  
    }catch(Exception ex){  
        System.out.println(ex.getMessage());  
    }
```

```
    }  
  
    return productsList;  
}  
  
public static ArrayList<ProductsD> FindProductsD(String  
ValToSearch)  
{  
    ArrayList<ProductsD> productsList = new  
ArrayList<ProductsD>();  
  
    ResultSet rs;  
  
    try{  
        Statement stmt = MySqlConnection.dbConnector().createStatement();  
        String searchQuery = "SELECT * FROM `products` WHERE  
(`ID` = '"+ValToSearch+"' )";  
        rs = stmt.executeQuery(searchQuery);  
  
        ProductsD product;  
  
        while(rs.next())  
        {  
            product = new ProductsD(  
                rs.getInt("ID"),  
                rs.getString("Name"),  
                rs.getString("ProdCode"),  
                rs.getString("Description"),  
                rs.getDouble("Price"),  
                rs.getInt("Tax"),  
                rs.getBoolean("Sales"),  
                rs.getBoolean("Stock"),  
                rs.getString("DateCreated"),  
                rs.getString("DateEdited")  
  
                );  
            productsList.add(product);  
        }  
  
    }catch(Exception ex){  
        System.out.println(ex.getMessage());  
    }  
  
    return productsList;  
}
```

```
}
```

## 5.5 Πακέτο products

### 5.5.1 InvoiceG.java

```
package invoice;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.ListSelectionModel;
import javax.swing.border.EmptyBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.DefaultTableModel;

import contacts.ContactPG;
import contacts.ContactsD;
import products.NewProductG;
import products.ProductsD;
import java.awt.Font;
import java.awt.Color;

public class InvoiceG extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTable table;
    private JButton btnNewButton_2;
```



```
public InvoiceG() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 681, 419);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("ΤΙΜΟΛΟΓΙΑ");
    lblNewLabel.setBackground(new Color(152, 251, 152));
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel.setOpaque(true);
    lblNewLabel.setBounds(0, 0, 665, 43);
    contentPane.add(lblNewLabel);

    JButton btnnewinvoice = new JButton("NEO ΤΙΜΟΛΟΓΙΟ");
    btnnewinvoice.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            NewInvoice newinvoice = new NewInvoice();
            newinvoice.setVisible(true);

        }
    });
    btnnewinvoice.setBounds(10, 54, 153, 33);
    contentPane.add(btnnewinvoice);

    textField = new JTextField();
    textField.addKeyListener(new KeyAdapter() {
        @Override
        public void keyTyped(KeyEvent e) {
            if(textField.getText().length() >= 0)
            {
                try {

                    findInvoice();

                } catch (SQLException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            }else {
                table.setModel(new DefaultTableModel(
                    new Object[][] {
                    },
                    new String[] {
```

```

        "ID", "ΕΠΩΝΥΜΙΑ", "ΑΦΜ", "ΗΜ.ΕΚΔΟΣΗΣ",
"ΠΟΣΟ", "ΠΛΗΡΩΜΕΝΟ"
        }
    ));

        }
    });

    textField.setBounds(207, 54, 256, 33);
    contentPane.add(textField);
    textField.setColumns(10);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(0, 113, 665, 229);
    contentPane.add(scrollPane);

    table = new JTable();
    table.setRowHeight(20);

    table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    table.setModel(new DefaultTableModel(
        new Object[][] {
        },
        new String[] {
            "ID", "ΕΠΩΝΥΜΙΑ", "ΑΦΜ", "ΗΜ.ΕΚΔΟΣΗΣ",
"ΠΟΣΟ", "ΚΑΤΑΣΤΑΣΗ"
        }
    ));
    scrollPane.setViewportView(table);

    JButton btnNewButton = new JButton("ΕΜΦΑΝΙΣΗ
ΤΙΜΟΛΟΓΙΟΥ");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            openInvoice();

        }
    });

    btnNewButton.setBounds(0, 346, 190, 34);
    contentPane.add(btnNewButton);
    btnNewButton.setEnabled(false);

    JButton btnNewButton_1 = new JButton("ΕΠΙΣΤΡΟΦΗ");

```

```
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        dispose();
    }
});
btnNewButton_1.setBounds(548, 347, 117, 33);
contentPane.add(btnNewButton_1);

btnNewButton_2 = new JButton("ΠΛΗΡΩΜΗ");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        payment();

    }
});
btnNewButton_2.setBounds(200, 346, 117, 34);
contentPane.add(btnNewButton_2);
btnNewButton_2.setEnabled(false);

table.getSelectionModel().addListSelectionListener(new
ListSelectionListener(){
    public void valueChanged(ListSelectionEvent e) {
        //enable button - put it in an EDT to be safe
though
        ListSelectionModel lsm =
(ListSelectionModel)e.getSource();
        if (lsm.isSelectionEmpty())
        {
            btnNewButton.setEnabled(false);
            btnNewButton_2.setEnabled(false);
        }
        else
        {
            btnNewButton.setEnabled(true);
            int row = table.getSelectedRow();
            String value =
table.getModel().getValueAt(row, 5).toString();

            if(value=="true") {

                btnNewButton_2.setEnabled(true);
                btnNewButton_2.setText("ΠΛΗΡΩΜΕΝΟ");
            }else {
                btnNewButton_2.setEnabled(true);
                btnNewButton_2.setText("ΠΛΗΡΩΜΗ");
            }
        }
    }
});
```

```
        }
    });

    fillTable();
}

public void fillTable() {

    ArrayList<InvoiceD> invoices = InvoiceD.ListInvoices();

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"ID",
"ΕΠΩΝΥΜΙΑ", "ΑΦΜ", "ΗΜ.ΕΚΔΟΣΗΣ", "ΠΟΣΟ", "ΠΛΗΡΩΜΕΝΟ"});
    // Object[] row = new Object[6];

    for(int i = 0; i < invoices.size(); i++)
    {
        Object[] row = new Object[6];
        ArrayList<ContactsD> contact =
ContactsD.FindContact(Integer.toString(invoices.get(i).getClientId
()));

        row[0] = invoices.get(i).getId();
        row[1] = contact.get(0).getCname();
        row[2] = contact.get(0).getAfm();
        row[3] = invoices.get(i).getInvoiceDate();
        row[4] = invoices.get(i).getTotal();

        if(invoices.get(i).getPaidStatus()) {
            row[5]="NAI";
        }else {
            row[5]="OXI";
        }

        //row[5] = invoices.get(i).getPaidStatus();
        model.addRow(row);
    }
    table.setModel(model);
}
```

```
public void findInvoice() throws SQLException{

    ArrayList<InvoiceD> invoices =
InvoiceD.FindInvoice(textField.getText());

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"ID",
"ΕΠΩΝΥΜΙΑ", "ΑΦΜ", "ΗΜ.ΕΚΔΟΣΗΣ", "ΠΟΣΟ", "ΠΛΗΡΩΜΕΝΟ"});
    Object[] row = new Object[6];

    for(int i = 0; i < invoices.size(); i++)
    {
        ArrayList<ContactsD> contact =
ContactsD.FindContact(Integer.toString(invoices.get(i).getClientId
()));

        row[0] = invoices.get(i).getId();
        row[1] = contact.get(0).getCname();
        row[2] = contact.get(0).getAfm();
        row[3] = invoices.get(i).getInvoiceDate();
        row[4] = invoices.get(i).getTotal();
        row[5] = invoices.get(i).getPaidStatus();
        model.addRow(row);
    }
    table.setModel(model);

}
```

```
public void openInvoice() {

    int column = 0;
    int row = table.getSelectedRow();
    String value = table.getModel().getValueAt(row,
column).toString();

    //System.out.print(value);
    ArrayList<InvoiceD> invoice =
InvoiceD.SelectInvoice(value);

    InvoicePG invoicepg = new InvoicePG();
    invoicepg.setVisible(true);
    invoicepg.PreviewInvoice(invoice);

}
```



```
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import javax.swing.JSeparator;
import java.awt.Color;
import java.awt.Container;

import com.toedter.calendar.JDateChooser;

import contacts.ContactsD;
import contacts.Epafes;
import products.ProductsD;
import products.ProductsG;

import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.math.RoundingMode;
import java.text.DateFormat;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.awt.event.ActionEvent;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JFormattedTextField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;
import javax.swing.event.ChangeListener;
import javax.swing.event.ChangeEvent;
import javax.swing.SwingConstants;
import java.awt.Font;

public class NewInvoice extends JFrame {

    private JPanel contentPane;
    private JTable table;
    public JButton btnclient;
    public ArrayList<ContactsD> contact = null;
    public JComboBox comboBox;
    private JTextField textField_1;
```

```
public ArrayList<ProductsD> product =null;
public JSpinner txtp;
public JComboBox comboBox_1;
private JLabel lbla;
private JLabel lblat;
private JSpinner spinner;
private DefaultTableModel model;
private JLabel lblfinal;
private JLabel lblffinal;
private JLabel label_6;
private JLabel label_5;
private JLabel lblNewLabel_4;

public double f24 , f13 , f6;
public double ff, f;

private Object[] row;
private Object[][] tablecell=new Object[10][6];
private int i;
private Object forlistener="False";
private JButton btnNewButton_1;
private JDateChooser dateChooser;
private JLabel lblar;
private JFormattedTextField ftxt;
private JButton btnNewButton_3;
private InvoiceD invoice;
private JLabel label_7;
private JLabel label_9;
private JLabel label_10;

public NewInvoice() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 853, 494);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("NEO ΤΙΜΟΛΟΓΙΟ");
    lblNewLabel.setBackground(new Color(152, 251, 152));
    lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
    lblNewLabel.setOpaque(true);
    lblNewLabel.setBounds(0, 0, 858, 33);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("ΠΕΛΑΤΗΣ*");
    lblNewLabel_1.setBounds(10, 40, 65, 14);
    contentPane.add(lblNewLabel_1);
```



```
        btnclient = new JButton("ΕΠΙΛΟΓΗ ΠΕΛΑΤΗ");
        btnclient.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                contactselect();

            }
        });
        btnclient.setBounds(95, 36, 272, 23);
        contentPane.add(btnclient);

        dateChooser = new JDateChooser();
        dateChooser.setDateFormatString("dd/MM/yyyy");

        Date date = new Date();
        dateChooser.setDate(date);

        dateChooser.getCalendarButton().addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
            }
        });
        dateChooser.setBounds(454, 40, 159, 20);
        contentPane.add(dateChooser);

        JLabel lblNewLabel_2 = new JLabel("ΗΜΕΡ*");
        lblNewLabel_2.setBounds(404, 40, 46, 14);
        contentPane.add(lblNewLabel_2);

        JLabel lblNewLabel_3 = new JLabel("Α.ΠΑΡΑΣΤ*");
        lblNewLabel_3.setBounds(633, 44, 65, 14);
        contentPane.add(lblNewLabel_3);

        JLabel label = new JLabel("ΠΡΟΙΟΝΤΑ");
        label.setBounds(0, 65, 75, 14);
        contentPane.add(label);

        JLabel label_1 = new JLabel("Τιτλος");
        label_1.setBounds(10, 101, 46, 14);
        contentPane.add(label_1);

        NumberFormat format = DecimalFormat.getInstance();
        format.setMinimumFractionDigits(2);
        format.setMaximumFractionDigits(2);
        format.setRoundingMode(RoundingMode.HALF_UP);

        txttp =new JSpinner();
```

```

        txtp.setBounds(394, 117, 80, 20);
        txtp.setModel(new SpinnerNumberModel(new Double(1), new
Double(0), null, new Double(1)));
        contentPane.add(txtp);

        comboBox_1 = new JComboBox();
        comboBox_1.setModel(new DefaultComboBoxModel(new
String[] {"24", "13", "6"}));
        comboBox_1.setBounds(495, 117, 46, 20);
        contentPane.add(comboBox_1);

        btnNewButton_1 = new JButton("ΠΡΟΣΘΗΚΗ");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                if(product != null) {

                    addToTable();

                }else {
                    JOptionPane.showMessageDialog(null,
"ΠΑΡΑΚΑΛΩ ΕΠΙΛΕΞΤΕ ΠΡΟΙΟΝ", "InfoBox: " + "ΠΡΟΣΟΧΗ",
JOptionPane.INFORMATION_MESSAGE);
                }

            }
        });
        btnNewButton_1.setBounds(573, 116, 104, 23);
        contentPane.add(btnNewButton_1);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(12, 167, 705, 194);
        contentPane.add(scrollPane);

        table = new JTable();
        table.setModel(new DefaultTableModel(
            new Object[][] {
            },
            new String[] {
                "ΠΡΟΙΟΝ", "ΠΟΣΟΤΗΤΑ", "ΤΙΜΗ", "ΦΠΑ",
"ΣΥΝΟΛΟ"
            }
        ));
        model = new DefaultTableModel();
        model.setColumnIdentifiers(new Object[]{"ΠΡΟΙΟΝ",
"ΠΟΣΟΤΗΤΑ", "ΤΙΜΗ", "ΦΠΑ", "ΣΥΝΟΛΟ"});
        scrollPane.setViewportViewView(table);
    
```

```
textField_1 = new JTextField();
textField_1.setEditable(false);
textField_1.setBounds(10, 117, 239, 20);
contentPane.add(textField_1);
textField_1.setColumns(10);

JButton btnNewButton = new JButton("ΕΠΙΛΟΓΗ
ΠΡΟΙΟΝΤΟΣ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        productselect();
    }
});
btnNewButton.setBounds(10, 138, 141, 23);
contentPane.add(btnNewButton);

lblat = new JLabel("Δαθες.Αποθ :");
lblat.setBounds(161, 142, 88, 14);
lblat.setVisible(false);
contentPane.add(lblat);

lbla = new JLabel("");
lbla.setBounds(255, 142, 46, 14);
lbla.setVisible(false);
contentPane.add(lbla);

spinner = new JSpinner();
spinner.addChangeListener(new ChangeListener() {
    public void stateChanged(ChangeEvent e) {

        if (forlistener=="False")
        {
            available_quantity_change();
        }else
        {
            lbla.setText(null);
            forlistener="False";
        }
    }
});
spinner.setModel(new SpinnerNumberModel(new Double(1),
new Double(0), null, new Double(1)));
spinner.setBounds(285, 117, 86, 20);
contentPane.add(spinner);
```

```

JButton btnNewButton_2 = new JButton("ΑΠΟΘΗΚΕΥΣΗ");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        check();
    }
});
btnNewButton_2.setBounds(706, 412, 131, 43);
contentPane.add(btnNewButton_2);

JLabel label_2 = new JLabel("ΦΠΑ 24 :");
label_2.setBounds(372, 363, 59, 14);
contentPane.add(label_2);

JLabel label_3 = new JLabel("ΦΠΑ 13 :");
label_3.setBounds(372, 388, 59, 14);
contentPane.add(label_3);

JLabel label_4 = new JLabel(" ΦΠΑ 6 :");
label_4.setBounds(372, 412, 59, 14);
contentPane.add(label_4);

lblNewLabel_4 = new JLabel("");
lblNewLabel_4.setBounds(441, 363, 65, 14);
contentPane.add(lblNewLabel_4);

label_5 = new JLabel("");
label_5.setBounds(441, 388, 46, 14);
contentPane.add(label_5);

label_6 = new JLabel("");
label_6.setBounds(441, 412, 46, 14);
contentPane.add(label_6);

JLabel lblNewLabel_5 = new JLabel("ΣΥΝ. ΠΛΗΝ ΦΠΑ : ");
lblNewLabel_5.setBounds(527, 363, 104, 14);
contentPane.add(lblNewLabel_5);

lblffinal = new JLabel("");
lblffinal.setBounds(634, 363, 83, 14);
contentPane.add(lblffinal);

JLabel label_8 = new JLabel("ΤΕΛΙΚΟ ΣΥΝΟΛΟ : ");
label_8.setBounds(527, 398, 104, 14);
contentPane.add(label_8);

lblffinal = new JLabel("");
lblffinal.setBounds(644, 398, 73, 14);
contentPane.add(lblffinal);

```

```
        JButton button = new JButton("ΔΙΑΓΡΑΦΗ");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                deleteTableRow();

            }

        });
        button.setBounds(49, 384, 113, 23);
        contentPane.add(button);

        lblar = new JLabel("");
        lblar.setBounds(671, 77, 46, 14);
        contentPane.add(lblar);

        ftxt = new JFormattedTextField();
        ftxt.setBounds(708, 40, 75, 20);
        contentPane.add(ftxt);

        btnNewButton_3 = new JButton("ΑΚΥΡΩΣΗ");
        btnNewButton_3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dispose();
            }
        });
        btnNewButton_3.setBounds(706, 374, 131, 33);
        contentPane.add(btnNewButton_3);

        invoice = new Invoiced();
        invoice.last_inserted_id();
        ftxt.setValue(invoice.idToInsert());

        label_7 = new JLabel("Ποσοτητα");
        label_7.setBounds(284, 101, 83, 14);
        contentPane.add(label_7);

        label_9 = new JLabel("Τιμη");
        label_9.setBounds(394, 101, 56, 14);
        contentPane.add(label_9);

        label_10 = new JLabel("ΦΠΑ");
```

```
        label_10.setBounds(495, 101, 46, 14);
        contentPane.add(label_10);

    }

    public void contactselect() {
        Epafes epafes=new Epafes();

        epafes.btnselect.setVisible(true);
        epafes.setVisible(true);

        epafes.btnselect.addActionListener(new ActionListener()
        {

            public void actionPerformed(ActionEvent e) {

                int column = 0;
                int row = epafes.table.getSelectedRow();

                String value =
                epafes.table.getModel().getValueAt(row, column).toString();
                contact = ContactsD.FindContact(value);

                btnclient.setText(contact.get(0).getCname());

                epafes.dispose();

            }
        });
    }

    public void productselect() {
        ProductsG productsg = new ProductsG();

        productsg.btnok.setVisible(true);
        //productsg.btnok.setEnabled(false);
        productsg.setVisible(true);

        productsg.btnok.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e) {

    int column = 0;
    int row = productsg.table.getSelectedRow();

    String value =
productsg.table.getModel().getValueAt(row, column).toString();

    product = ProductsD.FindProductsD(value);

    if(product.get(0).getSales()) {

textField_1.setText(product.get(0).getName());
txtp.setValue(product.get(0).getPrice());

        if(product.get(0).getTax()==24) {
            comboBox_1.setSelectedIndex(0);
        }else if(product.get(0).getTax()==13) {
            comboBox_1.setSelectedIndex(1);
        }else {
            comboBox_1.setSelectedIndex(2);
        }

        if(product.get(0).getStock()) {
            lblat.setVisible(true);
            lbla.setVisible(true);

ProductsD.showInventory(product.get(0).getId());

            double t = product.get(0).getTotal();
            double y = (double)
spinner.getValue();

            lbla.setText(String.valueOf(t-y));

        }

productsg.dispose();
    }
    else {
        JOptionPane.showMessageDialog(null,
"ΤΟ ΠΡΟΙΟΝ ΕΙΝΑΙ ΑΝΕΝΕΡΓΟ ΔΕΝ ΜΠΟΡΕΙ ΝΑ ΓΙΝΕΙ ΧΡΗΣΗ ΤΟΥ",
"ΠΡΟΣΟΧΗ", JOptionPane.INFORMATION_MESSAGE);
    }
}
});
```

```
 }//telos product select

public void available_quantity_change() {

    //allazei tin diathesimh posothta sto label
    double t = product.get(0).getTotal();
    double y = (double)spinner.getValue();

    lbla.setText(String.valueOf(t-y));

}

public void quantity_change() {

    //allazei tin diathesimh posothta sthn vash afou
    patitheï to add

    double t = product.get(0).getTotal();
    double y = (double) tablecell[i][1];

    double u = t-y;

    ProductsD.updateInventory(u,
product.get(0).getId());

}

public void addToTable() {

    double y = (double) spinner.getValue();
    double p = (Double) txtp.getValue();

    //APOTHIKEYSH SE ENAN PINAKA
    row = new Object[6];
    i = table.getRowCount();

    System.out.println(i);
    // row[0] = product.get(i).getId();
    row[0] = textField_1.getText(); //proion
    row[1] = spinner.getValue();
    row[2] = txtp.getValue();
}
```



```
        row[3] = comboBox_1.getSelectedItemAt();
        row[4] = y*p;

        model.addRow(row);
        table.setModel(model);

        row[5] = product.get(0).getId();

        tablecell[i]=row;

        updateFinalPrice();

        if(product.get(0).getStock()) {
            quantity_change();
        }

        setnull();

    }

    public void setnull() {

        textField_1.setText(null);
        forlistener = "True";
        spinner.setValue(1.0);
        txtp.setValue(1.0);
        comboBox_1.setSelectedItemAt(0);
        lbla.setText(null);
        product =null;
    }

    public void updateFinalPrice(){

        double q= (double)row[1];
        double w =(double) row[2];
        double e =(double)row[4];    //meriko sunolo

        DecimalFormat df = new DecimalFormat("#.##");

        if(row[3].toString()== "24" ) {

            f24+= Double.valueOf(df.format( (q*w) * 24.0f /
100));

            lblNewLabel_4.setText(Double.toString(f24));
```

```
    }else if(row[3].toString()== "13") {  
        f13+= Double.valueOf(df.format((q*w) * 13.0f /  
100));  
        label_5.setText(Double.toString(f13));  
    }else {  
        f6+= Double.valueOf(df.format((q*w) * 6.0f /  
100));  
        label_6.setText(Double.toString(f6));  
    }  
    ff+= Double.valueOf(df.format(e)); //meriko  
    f= Double.valueOf(df.format(ff+f24+f13+f6));  
    lblffinal.setText(Double.toString(ff));  
    lblfinal.setText(Double.toString(f));  
  
}
```

```
public void deleteTableRow(){  
  
    int selectedrow = table.getSelectedRow();  
    int gid = (int) tablecell[selectedrow][5];  
    int ggid =(int) tablecell[selectedrow][5];  
    model.removeRow(selectedrow);  
  
    //diagrafh apo telika apotelesmata  
  
    updateFinalPriceChange(selectedrow);  
  
    //allagh apothikis ean uparxei  
    product = ProductsD.FindProductsD(  
Integer.toString(gid));  
    if(product.get(0).getStock()) {  
  
        double t1 =product.get(0).getTotal();  
        double throwback = (double)  
tablecell[selectedrow][1];  
        double total = t1+throwback;  
  
        addStock(ggid,total);  
    }  
}
```

```
    }

    //διαγραφη απο pinaka tablecell
    removeArrayRow(selectedrow);

    System.out.println(tablecell[selectedrow][0]);

}

public void updateFinalPriceChange(int selectedrow){

    double q= (double)tablecell[selectedrow][1];
    double w =(double) tablecell[selectedrow][2];
    double e =(double)tablecell[selectedrow][4]; //meriko
sunolo

    DecimalFormat df = new DecimalFormat("#.##");

    if(tablecell[selectedrow][3].toString()== "24" ) {

        f24-= Double.valueOf(df.format( (q*w) * 24.0f /
100));

        lblNewLabel_4.setText(Double.toString(f24));

    }else if(tablecell[selectedrow][3].toString()== "13") {

        f13-= Double.valueOf(df.format((q*w) * 13.0f /
100));

        label_5.setText(Double.toString(f13));

    }else {

        f6-= Double.valueOf(df.format((q*w) * 6.0f /
100));

        label_6.setText(Double.toString(f6));

    }

    ff-= Double.valueOf(df.format(e)); //meriko
//    DecimalFormat df = new DecimalFormat("#.##");
```

```
f= Double.valueOf(df.format(ff+f24+f13+f6));
```

```
lblffinal.setText(Double.toString(ff));
```

```
lblfinal.setText(Double.toString(f));
```

```
}
```

```
public void addStock(int gid,double total) {
```

```
ProductsD.updateInventory(total, gid);
```

```
}
```

```
public void removeArrayRow(int selectedrow) {
```

```
Object[][] tablecell2=new Object[tablecell.length-1][6];
```

```
System.arraycopy(tablecell, 0, tablecell2, 0,  
selectedrow);
```

```
System.arraycopy(tablecell, selectedrow+1, tablecell2,  
selectedrow, tablecell.length-selectedrow-1);
```

```
tablecell=tablecell2;
```

```
}
```

```
public void check() {
```

```
if(contact != null) {
```

```
invoice.addInvoiceToDB(contact.get(0).getId(),  
dateChooser.getDate(),f);
```

```
invoice.addProductsToDB(tablecell);
```

```
        ArrayList<InvoiceD> inv =
InvoiceD.SelectInvoice(ftxt.getText());

        InvoicePG invoicepg = new InvoicePG();
        invoicepg.PreviewInvoice(inv);
        invoicepg.generateInvoice();
        dispose();

    }else {
        JOptionPane.showMessageDialog(null, "ΠΑΡΑΚΑΛΩ
ΕΠΙΛΕΞΤΕ ΠΕΛΑΤΗ", "InfoBox: " + "ΠΡΟΣΟΧΗ",
JOptionPane.INFORMATION_MESSAGE);
    }

}

} //end class
```

### 5.5.3 InvoicePG.java

```
package invoice;

import java.awt.BorderLayout;
import java.awt.Desktop;
import java.awt.EventQueue;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Arrays;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;

import contacts.ContactsD;
import settings.Email;
import settings.GenerateInvoice;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.print.attribute.HashPrintRequestAttributeSet;
```

```
import javax.print.attribute.PrintRequestAttributeSet;
import javax.print.attribute.standard.Copies;
import javax.print.attribute.standard.MediaSize;
import javax.print.attribute.standard.Sides;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.SwingConstants;
import java.awt.event.ActionListener;
import java.awt.print.PrinterException;
import java.awt.print.PrinterJob;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.awt.event.ActionEvent;
import javax.print.*;
import java.awt.Color;
import java.awt.Font;

public class InvoicePG extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private DefaultTableModel model;
    private ArrayList<InvoiceD> invoice;
    private ArrayList<ContactsD> contact;
    private JLabel lblNewLabel_2;
    private JLabel label_1;
    private JLabel lblNewLabel_3;
    private JLabel lblNewLabel_4;
    private ArrayList<InvoiceD> invoiceitems;

    double f6,f13,f24,ff,f;
    private JLabel label_9;
    private JLabel label_8;
    private JLabel label_7;
    private JLabel label_11;
    private JLabel label_12;

    public InvoicePG() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 689, 464);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
}
```

```
JLabel lblNewLabel = new JLabel("ΤΙΜΟΛΟΓΙΟ");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
lblNewLabel.setBackground(new Color(152, 251, 152));
lblNewLabel.setOpaque(true);
lblNewLabel.setBounds(0, 0, 673, 38);
contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("ΕΠΩΝΥΜΙΑ :");
lblNewLabel_1.setBounds(10, 40, 85, 14);
contentPane.add(lblNewLabel_1);

    lblNewLabel_2 = new JLabel("");
lblNewLabel_2.setBounds(90, 40, 188, 14);
contentPane.add(lblNewLabel_2);

JLabel label = new JLabel("ΑΦΜ :");
label.setBounds(300, 40, 68, 14);
contentPane.add(label);

    label_1 = new JLabel("");
label_1.setBounds(378, 40, 108, 14);
contentPane.add(label_1);

JLabel label_2 = new JLabel("ΗΜ.ΕΚΔΟΣΗΣ :");
label_2.setBounds(12, 65, 83, 14);
contentPane.add(label_2);

    lblNewLabel_3 = new JLabel("");
lblNewLabel_3.setBounds(100, 65, 122, 14);
contentPane.add(lblNewLabel_3);

JLabel label_3 = new JLabel("ΑΡ.ΠΑΡΑΣΤΑΤΙΚΟΥ :");
label_3.setBounds(302, 65, 108, 14);
contentPane.add(label_3);

    lblNewLabel_4 = new JLabel("");
lblNewLabel_4.setBounds(420, 65, 108, 14);
contentPane.add(lblNewLabel_4);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(0, 108, 673, 194);
contentPane.add(scrollPane);

table = new JTable();
table.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
```

```
        "ΠΡΟΙΟΝ", "ΠΟΣΟΤΗΤΑ", "ΤΙΜΗ", "ΦΠΑ",  
"ΣΥΝΟΛΟ"  
    }  
    ));  
    model = new DefaultTableModel();  
    model.setColumnIdentifiers(new Object[]{"ΠΡΟΙΟΝ",  
"ΠΟΣΟΤΗΤΑ", "ΤΙΜΗ", "ΦΠΑ", "ΣΥΝΟΛΟ"});  
    scrollPane.setViewportView(table);  
  
    JLabel label_4 = new JLabel("ΦΠΑ 24 :");  
    label_4.setBounds(317, 307, 68, 14);  
    contentPane.add(label_4);  
  
    JLabel label_5 = new JLabel("ΦΠΑ 13 :");  
    label_5.setBounds(317, 332, 68, 14);  
    contentPane.add(label_5);  
  
    JLabel label_6 = new JLabel(" ΦΠΑ 6 :");  
    label_6.setBounds(317, 356, 68, 14);  
    contentPane.add(label_6);  
  
    label_7 = new JLabel("");  
    label_7.setBounds(408, 356, 58, 14);  
    contentPane.add(label_7);  
  
    label_8 = new JLabel("");  
    label_8.setBounds(408, 332, 65, 14);  
    contentPane.add(label_8);  
  
    label_9 = new JLabel("");  
    label_9.setBounds(405, 307, 68, 14);  
    contentPane.add(label_9);  
  
    JLabel label_10 = new JLabel("ΣΥΝ. ΠΛΗΝ ΦΠΑ : ");  
    label_10.setBounds(474, 313, 108, 14);  
    contentPane.add(label_10);  
  
    label_11 = new JLabel("");  
    label_11.setBounds(579, 315, 92, 14);  
    contentPane.add(label_11);  
  
    label_12 = new JLabel("");  
    label_12.setBounds(576, 347, 88, 14);  
    contentPane.add(label_12);  
  
    JLabel label_13 = new JLabel("ΤΕΛΙΚΟ ΣΥΝΟΛΟ : ");  
    label_13.setBounds(474, 348, 199, 14);  
    contentPane.add(label_13);
```



```
        JButton btnNewButton = new JButton("ΑΠΟΘΗΚΕΥΣΗ");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                savePdfToFile();

            }
        });
        btnNewButton.setBounds(6, 313, 116, 57);
        contentPane.add(btnNewButton);

        JButton button = new JButton("ΑΠΟΣΤΟΛΗ");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                InvoiceD invo = new InvoiceD();
                invo.getPdfFromDb(invoice.get(0).getId());

                if(contact.get(0).getEmail()!="") {
                    Email mail =new
Email(contact.get(0).getEmail(),"ΤΙΜΟΛΟΓΙΟ" ,"ΤΙΜΟΛΟΓΙΟ ΜΕ ΑΡΙΘΜΟ
ΠΑΡΑΣΤΑΤΙΚΟΥ "+invo.getId(),invo.getPdf());

                    button.setText("ΣΤΑΛΘΗΚΕ");
                    button.setEnabled(false);
                }else {
                    JOptionPane.showMessageDialog(null,
"ΔΕΝ ΥΠΑΡΧΕΙ ΔΙΕΥΘΥΝΣΗ EMAIL ΣΤΗΝ ΕΠΑΦΗ", "InfoBox: " + "ΠΡΟΣΟΧΗ",
JOptionPane.INFORMATION_MESSAGE);
                }

            }

        });
        button.setBounds(132, 313, 116, 57);
        contentPane.add(button);

        JButton button_1 = new JButton("ΑΝΟΙΓΜΑ");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                opedPdf();

            }
        });
```

```
button_1.setBounds(9, 375, 116, 48);
contentPane.add(button_1);

JButton button_2 = new JButton("ΕΠΙΣΤΡΟΦΗ");
button_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
button_2.setBounds(556, 390, 117, 33);
contentPane.add(button_2);
}

public void PreviewInvoice(ArrayList<InvoiceD> invoice) {

    double q,w,e,t;
    this.invoice=invoice;

    int value = this.invoice.get(0).getClientId();

    contact =
ContactsD.FindContact(Integer.toString(value));

    lblNewLabel_2.setText(contact.get(0).getCname());

    label_1.setText(contact.get(0).getAfm());

    lblNewLabel_3.setText(invoice.get(0).getInvoiceDate());

    lblNewLabel_4.setText(Integer.toString(invoice.get(0).getId(
)));

    invoiceitems
=InvoiceD.InvoiceItems(Integer.toString(invoice.get(0).getId()));

    DecimalFormat df = new DecimalFormat("#.##");

    Object[] row = new Object[5];
    for(int i = 0; i < invoiceitems.size(); i++)
    {
        row[0] = invoiceitems.get(i).getpName();
        row[1] = invoiceitems.get(i).getpQuantity();
        row[2] = invoiceitems.get(i).getpPrice();
        row[3] = invoiceitems.get(i).getpTax();
        row[4] = invoiceitems.get(i).getpQuantity() *
invoiceitems.get(i).getpPrice();
    }
}
```

```
        model.addRow(row);

        q= invoiceitems.get(i).getpQuantity();
        w= invoiceitems.get(i).getpPrice();
        e= invoiceitems.get(i).getpQuantity() *
invoiceitems.get(i).getpPrice();
        t=invoiceitems.get(i).getpTax();

        if(t == 24 ) {

            f24+= Double.valueOf(df.format( (q*w) *
24.0f / 100));

            label_9.setText(Double.toString(f24));

        }else if(t == 13) {

            f13+= Double.valueOf(df.format((q*w) *
13.0f / 100));

            label_8.setText(Double.toString(f13));

        }else {

            f6+= Double.valueOf(df.format((q*w) * 6.0f /
100));

            label_7.setText(Double.toString(f6));
        }

        ff+= Double.valueOf(df.format(e)); //meriko

        f= Double.valueOf(df.format(ff+f24+f13+f6));
//teliko sunolo

        label_11.setText(Double.toString(ff));
        label_12.setText(Double.toString(f));

    }

    table.setModel(model);

}

public void generateInvoice() {

    GenerateInvoice pdf=new
GenerateInvoice(f,ff,f24,f13,f6,invoiceitems,contact,invoice);
    pdf.genpdf();
}
```

```
}
```

```
public void savePdfToFile() {  
  
    FileOutputStream fos = null;  
    try {  
  
        JFileChooser fileChooser = new JFileChooser();  
        fileChooser.setDialogTitle("ΑΠΟΘΗΚΕΥΣΗ PDF");  
  
        int selectedFile =  
fileChooser.showSaveDialog(null);  
  
        if (selectedFile == JFileChooser.APPROVE_OPTION)  
{  
  
            File pdftosave = fileChooser.getSelectedFile();  
            fos = new  
FileOutputStream(pdftosave.getAbsolutePath()+".pdf");  
  
            InvoiceD invo = new InvoiceD();  
            invo.getPdfFromDb(invoice.get(0).getId());  
            fos.write(invo.getPdf());  
            fos.flush();  
            fos.close();  
        }  
  
        } catch (FileNotFoundException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        } catch (IOException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
    }  
}
```

```
public void opedPdf() {  
  
    InvoiceD invo = new InvoiceD();  
    invo.getPdfFromDb(invoice.get(0).getId());
```

```
        if (Desktop.isDesktopSupported()) {
            try {

                FileOutputStream fos = null;

                File file =File.createTempFile("tempfile",
".pdf");

                fos = new FileOutputStream(file);

                fos.write(invo.getPdf());
                fos.flush();
                fos.close();

                Desktop.getDesktop().open(file);

                file.deleteOnExit();

            } catch (IOException ex) {
                // no application registered for PDFs
            }
        }
    }
}
```

}//telos

#### 5.5.4 InvoiceD.java

```
package invoice;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.swing.*;

import com.mysql.jdbc.PreparedStatement;

import contacts.ContactsD;
import products.ProductsD;
import util.MySqlConnection;

public class InvoiceD {

    private DateFormat dateFormat = new SimpleDateFormat("yyyy-
MM-dd");
    private static DateFormat dateFormat2 = new
SimpleDateFormat("yyyy-MM-dd");
    private int lastid;
    private Date dateinserted;
    private int clientid;
    private int id;
    private String invoicedate;
    private String status;
    private Double total;
    private boolean paidstatus;
    private String pname;
    private int pid;
    private Double pquantity;
    private Double pprice;
    private Double ptax;
    private byte[] pdf;

    public InvoiceD(int ID,int ClientID,String
InvoiceDate,String Status,double Total,boolean PaidStatus)
    {
        this.id = ID;
        this.clientid = ClientID;
        this.invoicedate =InvoiceDate ;
        this.status = Status;
        this.total = Total;
        this.paidstatus = PaidStatus;
    }
}
```

```
    }

    public InvoiceD(int pID,String pName,double
pQuantity,double pPrice,double pTax)
    {
        this.pid = pID;
        this.pname = pName;
        this.pquantity =pQuantity ;
        this.pprice = pPrice;
        this.ptax = pTax;

    }

    public InvoiceD() {
    }

    int lastInsertedId() {
        return lastid;
    }

    int idToInsert() {
        return lastid+1;
    }

    public int getId()
    {
        return id;
    }

    public int getClientId()
    {
        return clientid;
    }
}
```

```
public String getInvoiceDate()
{
    return invoicedate;
}

public String getStatus()
{
    return status;
}

public Double getTotal()
{
    return total;
}

public boolean getPaidStatus()
{
    return paidstatus;
}

//items

public int getpId()
{
    return pid;
}

public String getpName()
{
    return pname;
}

public Double getpQuantity()
{
    return pquantity;
}

public Double getpPrice()
{
    return pprice;
}

public Double getpTax()
{
    return ptax;
}
```



```
    }

    public byte[] getPdf()
    {
        return pdf;
    }

    public void last_inserted_id() {
        ResultSet rs;

        try {
            Statement stmt =
                MySqlConnection.dbConnector().createStatement();

            String sql = "SELECT id FROM invoices ORDER BY
id DESC LIMIT 1";

            rs = stmt.executeQuery(sql);

            while(rs.next())
            {

                lastid = rs.getInt("ID");

            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    public void addInvoiceToDB(int contactid, Date date, double
f) {

        //DateFormat dateFormat = new SimpleDateFormat("yyyy-
MM-dd");
        dateinserted =date;
        try {
```

```
Statement stmt =
MySqlConnection.dbConnector().createStatement();

String sql = "INSERT INTO
invoices(ID,ClientID,InvoiceDate,Status,Total,PaidStatus) VALUES
("
    + "NULL" + ", '" +
    contactid + "', '" +
    dateFormat.format(date) + "', '" +
    "issued" + "', '" +
    f + "', '" +
    "N" + "' )";

    stmt.executeUpdate(sql);

} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}

public void addProductsToDB(Object[][] tablecell) {

int id =lastid+1;

try {

Statement stmt =
MySqlConnection.dbConnector().createStatement();

for (int i=0;tablecell[i][0]!=null;i++) {

String sql = "INSERT INTO
invoiceitems(InvoiceID,ProductID,ProductName,Quantity,Price,Tax,DateInserted) VALUES ("
    + id + ", '" +
    tablecell[i][5] + "', '" +
    tablecell[i][0] + "', '" +
    tablecell[i][1] + "', '" +
    tablecell[i][2] + "', '" +
    tablecell[i][3] + "', '" +
    dateFormat.format(dateinserted) + "' )";
```

```
                stmt.executeUpdate(sql);
            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

public static ArrayList<InvoiceD> ListInvoices()
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoices` ORDER BY
`ID` DESC LIMIT 20";
        rs = stmt.executeQuery(searchQuery);

        InvoiceD invoice;

        while(rs.next())
        {
            invoice = new InvoiceD(
                rs.getInt("ID"),
                rs.getInt("ClientID"),
                rs.getString("InvoiceDate"),
                rs.getString("Status"),
                rs.getDouble("Total"),
                rs.getBoolean("PaidStatus")
            );
            invoiceList.add(invoice);
        }

    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }

    return invoiceList;
}
```

```
}
```

```
public static ArrayList<InvoiceD> FindInvoice(String ValToSearch)
throws SQLException
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

    ResultSet rs=null;

    try{
        Statement stmt = MySqlConn.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoices` INNER JOIN
`contacts` ON invoices.ClientID = contacts.ID WHERE (CompanyName
LIKE '%" + ValToSearch + "%' OR VatNumber LIKE '%" + ValToSearch + "%')";
        rs = stmt.executeQuery(searchQuery);
        InvoiceD invoice;

        while(rs.next())
        {
            invoice = new InvoiceD(
                rs.getInt("invoices.ID"),
                rs.getInt("ClientID"),
                rs.getString("InvoiceDate"),
                rs.getString("Status"),
                rs.getDouble("Total"),
                rs.getBoolean("PaidStatus")
            );
            invoiceList.add(invoice);
        }

    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }

    return invoiceList;
}
```

```
public static ArrayList<InvoiceD> SelectInvoice(String
ValToSearch)
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();
```

```

ResultSet rs;

try{
    Statement stmt = MySqlConnection.dbConnector().createStatement();
    String searchQuery = "SELECT * FROM `invoices` WHERE
(`ID` = '"+ValToSearch+"' )";
    rs = stmt.executeQuery(searchQuery);

    Invoiced invoice;

    while(rs.next())
    {
        invoice = new Invoiced(
            rs.getInt("invoices.ID"),
            rs.getInt("ClientID"),
            rs.getString("InvoiceDate"),
            rs.getString("Status"),
            rs.getDouble("Total"),
            rs.getBoolean("PaidStatus")
        );
        invoiceList.add(invoice);
    }

}catch(Exception ex){
    System.out.println(ex.getMessage());
}

return invoiceList;
}

public static ArrayList<Invoiced> InvoiceItems(String ValToSearch)
{
    ArrayList<Invoiced> invoiceItems = new ArrayList<Invoiced>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoiceitems` WHERE
(`InvoiceID` = '"+ValToSearch+"' )";
        rs = stmt.executeQuery(searchQuery);

        Invoiced invoice;

        while(rs.next())
        {
            invoice = new Invoiced(

```

```
        rs.getInt("ProductID"),
        rs.getString("ProductName"),
        rs.getDouble("Quantity"),
        rs.getDouble("Price"),
        rs.getDouble("Tax")
    );
    invoiceItems.add(invoice);
}

}catch(Exception ex){
    System.out.println(ex.getMessage());
}

return invoiceItems;
}

public static void addPdfToDb(int i, ByteArrayOutputStream baos)
{

    InputStream is = new
    ByteArrayInputStream(baos.toByteArray());

    PreparedStatement stmt = null;
    try {

        //Statement stmt =
        MySqlConnection.dbConnector().createStatement();

        stmt = (PreparedStatement)
        MySqlConnection.dbConnector().prepareStatement("INSERT INTO invoicepdf
        VALUES(?,?)");

        stmt.setInt(1,i);
        stmt.setBinaryStream(2,is);
        stmt.executeUpdate();

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
    }

    public void getPdfFromDb(int value) {

        ResultSet rs;

        try{
            Statement stmt =
                MySqlConnection.dbConnector().createStatement();
            String searchQuery = "SELECT * FROM `invoicepdf`
                WHERE `InvoiceID`="+value;
            rs = stmt.executeQuery(searchQuery);

            //Blob blob;

            while(rs.next())
            {
                this.id=rs.getInt("InvoiceID");

                this.pdf=rs.getBytes("Pdf");

                System.out.println(this.pdf);

            }

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

    public static void changePayment(String value, String b) {

        PreparedStatement stmt = null;
        try {

            //Statement stmt =
            MySqlConnection.dbConnector().createStatement();
```

```
        stmt = (PreparedStatement)
        MySqlConnection.dbConnector().prepareStatement("UPDATE invoices SET
        PaidStatus=? WHERE ID="+value);
```

```
        stmt.setString(1,b);
        stmt.executeUpdate();
```

```
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
public static ArrayList<InvoiceD> selectUnpaidInvoices(int id2)
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoices` WHERE
        (`ClientID` = '"+id2+"' AND `PaidStatus`='N' )";
        rs = stmt.executeQuery(searchQuery);

        InvoiceD invoice;

        while(rs.next())
        {
            invoice = new InvoiceD(
                rs.getInt("invoices.ID"),
                rs.getInt("ClientID"),
                rs.getString("InvoiceDate"),
                rs.getString("Status"),
                rs.getDouble("Total"),
                rs.getBoolean("PaidStatus")
            );
            invoiceList.add(invoice);
        }
    }
```



```
}catch(Exception ex){
    System.out.println(ex.getMessage());
}

return invoiceList;
}

public static ArrayList<InvoiceD> contactsPaymDue(Date from,Date
to)
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoices` WHERE
(`PaidStatus`='N' AND `InvoiceDate` BETWEEN
CAST('"+dateFormat2.format(from)+"' AS DATE) AND
CAST('"+dateFormat2.format(to)+"' AS DATE))";
        rs = stmt.executeQuery(searchQuery);

        InvoiceD invoice;

        while(rs.next())
        {
            invoice = new InvoiceD(
                rs.getInt("invoices.ID"),
                rs.getInt("ClientID"),
                rs.getString("InvoiceDate"),
                rs.getString("Status"),
                rs.getDouble("Total"),
                rs.getBoolean("PaidStatus")
            );
            invoiceList.add(invoice);
        }

    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }

    return invoiceList;
}

public static ArrayList<InvoiceD> contactsBA(Date from,Date to)
{
```

```
ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

ResultSet rs;

try{
    Statement stmt = MySqlConnection.dbConnector().createStatement();
    String searchQuery = "SELECT * FROM `invoices` WHERE
`InvoiceDate` BETWEEN CAST('"+dateFormat2.format(from)+"' AS DATE)
AND CAST('"+dateFormat2.format(to)+"' AS DATE) ";
    rs = stmt.executeQuery(searchQuery);

    InvoiceD invoice;

    while(rs.next())
    {
        invoice = new InvoiceD(
            rs.getInt("invoices.ID"),
            rs.getInt("ClientID"),
            rs.getString("InvoiceDate"),
            rs.getString("Status"),
            rs.getDouble("Total"),
            rs.getBoolean("PaidStatus")
        );
        invoiceList.add(invoice);
    }

}catch(Exception ex){
    System.out.println(ex.getMessage());
}

return invoiceList;
}

public static ArrayList<InvoiceD> salesPPI(Date from,Date to)
{
    ArrayList<InvoiceD> invoiceList = new ArrayList<InvoiceD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConnection.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoices` WHERE
`InvoiceDate` BETWEEN CAST('"+dateFormat2.format(from)+"' AS DATE)
AND CAST('"+dateFormat2.format(to)+"' AS DATE) ";
        rs = stmt.executeQuery(searchQuery);

        InvoiceD invoice;

        while(rs.next())
```

```
{
    invoice = new InvoiceD(
        rs.getInt("invoices.ID"),
        rs.getInt("ClientID"),
        rs.getString("InvoiceDate"),
        rs.getString("Status"),
        rs.getDouble("Total"),
        rs.getBoolean("PaidStatus")
    );
    invoiceList.add(invoice);
}

}catch(Exception ex){
    System.out.println(ex.getMessage());
}

return invoiceList;
}

public static ArrayList<InvoiceD> prodSS(Date from,Date to)
{
    ArrayList<InvoiceD> invoiceitems = new ArrayList<InvoiceD>();

    ResultSet rs;

    try{
        Statement stmt = MySqlConn.dbConnector().createStatement();
        String searchQuery = "SELECT * FROM `invoiceitems` WHERE
`DateInserted` BETWEEN CAST('"+dateFormat2.format(from)+"' AS
DATE) AND CAST('"+dateFormat2.format(to)+"' AS DATE) ";
        rs = stmt.executeQuery(searchQuery);

        InvoiceD invoice;

        while(rs.next())
        {
            invoice = new InvoiceD(
                rs.getInt("ProductID"),
                rs.getString("ProductName"),
                rs.getDouble("Quantity"),
                rs.getDouble("Price"),
                rs.getDouble("Tax")

            );

            invoiceitems.add(invoice);
        }

    }catch(Exception ex){
```

```
        System.out.println(ex.getMessage());
    }

    return invoiceitems;
}

} //end Invoiced
```

## 5.6 Πακέτο reports

### 5.6.1 ReportsG.java

```
package reports;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

import com.toedter.calendar.JDateChooser;

import contacts.ContactsD;
import invoice.Invoiced;

import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JButton;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.ListSelectionModel;
import javax.swing.RowSorter;
import javax.swing.SortOrder;

import java.awt.event.ActionListener;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Date;
import java.awt.event.ActionEvent;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
```

```
import javax.swing.SwingConstants;

public class ReportsG extends JFrame {

    private JPanel contentPane;
    private JLabel lblNewLabel;
    private JPanel panelcontacts;
    private JDateChooser dateChooser;
    private JDateChooser dateChooser_1;
    private JTable table;
    private Date from;
    private Date to;
    private JLabel lblNewLabel_1;
    private JLabel label_3;
    private JPanel panelsales;
    private JLabel lblNewLabel_2;
    private JLabel label_4;
    private JLabel label_5;
    private JLabel label_6;
    private JLabel label_7;
    private JLabel lblNewLabel_10;
    private JPanel panelprod;
    private JTable table1;
    private JTable table2;
    private DecimalFormat df = new DecimalFormat("####0.00");

    public ReportsG() {

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 886, 495);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton btnNewButton = new JButton("ΕΠΑΦΗΣ");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                lblNewLabel.setText("ΑΝΑΦΟΡΕΣ ΕΠΑΦΩΝ");

                panelsales.setVisible(false);

                panelprod.setVisible(false);

                panelcontacts.setVisible(true);
            }
        });
    }
}
```

```
    }
  });
  btnNewButton.setBounds(0, 59, 146, 53);
  contentPane.add(btnNewButton);

  JButton button = new JButton("ΠΩΛΗΣΕΩΝ");
  button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

      lblNewLabel.setText("ΑΝΑΦΟΡΕΣ ΠΩΛΗΣΕΩΝ");

      panelcontacts.setVisible(false);

      panelprod.setVisible(false);

      panelsales.setVisible(true);
    }
  });
  button.setBounds(0, 113, 146, 53);
  contentPane.add(button);

  JButton button_1 = new JButton("ΠΡΟΙΟΝΤΩΝ");
  button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

      lblNewLabel.setText("ΑΝΑΦΟΡΕΣ ΠΡΟΙΟΝΤΩΝ");

      panelsales.setVisible(false);

      panelcontacts.setVisible(false);

      panelprod.setVisible(true);
    }
  });
  button_1.setBounds(0, 166, 146, 53);
  contentPane.add(button_1);

  lblNewLabel = new JLabel("ΑΝΑΦΟΡΕΣ");
  lblNewLabel.setBackground(new Color(192, 192, 192));
  lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
  lblNewLabel.setOpaque(true);
  lblNewLabel.setBounds(0, 0, 870, 59);
  contentPane.add(lblNewLabel);
```

```
        panel3();
        panel2();
        panel1();
        panelcontacts.setVisible(false);
        panelprod.setVisible(false);
        panelsales.setVisible(false);

        lblNewLabel.setText("ΑΝΑΦΟΡΕΣ");
    }

    public void panel1() {

        panelcontacts = new JPanel();
        panelcontacts.setBounds(145, 59, 725, 397);
        contentPane.add(panelcontacts);
        panelcontacts.setLayout(null);

        JLabel label = new JLabel("Τυπος Αναφορας");
        label.setFont(new Font("Tahoma", Font.PLAIN,
13));
        label.setBounds(10, 11, 137, 14);
        panelcontacts.add(label);

        JLabel label_1 = new JLabel("Απο");
        label_1.setFont(new Font("Tahoma", Font.PLAIN,
13));
        label_1.setBounds(243, 11, 137, 14);
        panelcontacts.add(label_1);

        JLabel label_2 = new JLabel("Εως");
        label_2.setFont(new Font("Tahoma", Font.PLAIN,
13));
        label_2.setBounds(456, 11, 137, 14);
        panelcontacts.add(label_2);

        JComboBox comboBox = new JComboBox();
        comboBox.setModel(new DefaultComboBoxModel(new
String[] {"ΥΠΟΛΟΙΠΟ ΕΠΑΦΩΝ", "ΜΕΓΑΛΥΤΕΡΟ ΠΟΣΟ"}));
        comboBox.setBounds(10, 36, 205, 20);
        panelcontacts.add(comboBox);

        Date date = new Date();
```

```
dateChooser = new JDateChooser();
dateChooser.setBounds(243, 36, 159, 20);
panelcontacts.add(dateChooser);
dateChooser.setDateFormatString("dd/MM/yyyy");
dateChooser.setDate(date);

dateChooser_1 = new JDateChooser();
dateChooser_1.setDateFormatString("dd/MM/yyyy");
dateChooser_1.setBounds(452, 36, 159, 20);
dateChooser_1.setDate(date);
panelcontacts.add(dateChooser_1);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setEnabled(true);
scrollPane.setBounds(10, 84, 705, 260);
panelcontacts.add(scrollPane);

table1 = new JTable();

table1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION)
;
        table1.setModel(new DefaultTableModel(
            new Object[][] {
            },
            new String[] {
                "ID ΠΕΛΑΤΗ", "ΕΠΩΝΥΜΙΑ",
                "ΑΝΕΞΟΦΛΗΤΑ ΤΙΜΟΛΟΓΙΑ", "ΣΥΝ.ΠΟΣΟ"
            }
        ));
        scrollPane.setViewportViewView(table1);

        JButton btnNewButton_1 = new
JButton("ΑΝΑΖΗΤΗΣΗ");
        btnNewButton_1.addActionListener(new
ActionListener() {
            public void
actionPerformed(ActionEvent e) {
                from= dateChooser.getDate();

                to= dateChooser_1.getDate();

                if(comboBox.getSelectedIndex()==0) { //ΥΠΟΛΟΙΠΟ ΕΠΑΦΩΝ
                    contactsPaymentDue();
                }

                if(comboBox.getSelectedIndex()==1) { //MEGALYTERO POSO
```



```
contactsBiggerAmount();
    }

    }
});
btnNewButton_1.setBounds(621, 35, 94, 23);
panelcontacts.add(btnNewButton_1);

lblNewLabel_1 = new JLabel("");
lblNewLabel_1.setBounds(400, 350, 94, 14);
panelcontacts.add(lblNewLabel_1);

label_3 = new JLabel("");
label_3.setBounds(553, 350, 94, 14);
panelcontacts.add(label_3);

}

public void panel2() {

    panelsales = new JPanel();
    panelsales.setBounds(145, 59, 725, 397);
    contentPane.add(panelsales);
    panelsales.setLayout(null);

    JLabel label = new JLabel("Τυπος Αναφορας");
    label.setFont(new Font("Tahoma", Font.PLAIN,
13));
    label.setBounds(10, 11, 137, 14);
    panelsales.add(label);

    JLabel label_1 = new JLabel("Απο");
    label_1.setFont(new Font("Tahoma", Font.PLAIN,
13));
    label_1.setBounds(243, 11, 137, 14);
    panelsales.add(label_1);
```

```

JLabel label_2 = new JLabel("Εως");
label_2.setFont(new Font("Tahoma", Font.PLAIN,
13));

label_2.setBounds(456, 11, 137, 14);
panelsales.add(label_2);

JComboBox comboBox = new JComboBox();
comboBox.setModel(new DefaultComboBoxModel(new
String[] {"ΠΑΡΑΣΤΑΤΙΚΑ ΠΩΛΗΣΕΩΝ"}));
comboBox.setBounds(10, 36, 205, 20);
panelsales.add(comboBox);

Date date = new Date();
dateChooser = new JDateChooser();
dateChooser.setBounds(243, 36, 159, 20);
panelsales.add(dateChooser);
dateChooser.setDateFormatString("dd/MM/yyyy");
dateChooser.setDate(date);

dateChooser_1 = new JDateChooser();
dateChooser_1.setDateFormatString("dd/MM/yyyy");
dateChooser_1.setBounds(452, 36, 159, 20);
dateChooser_1.setDate(date);
panelsales.add(dateChooser_1);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setEnabled(false);
scrollPane.setBounds(10, 84, 705, 260);
panelsales.add(scrollPane);

table2 = new JTable();

table2.setSelectionMode(ListSelectionModel.SINGLE_SELECTION)
;

        table2.setModel(new DefaultTableModel(
            new Object[][] {
            },
            new String[] {
                "ID ΤΙΜ", "ΦΠΑ 6", "ΦΠΑ 13", "ΦΠΑ
24", "ΚΑΘΑΡΗ ΑΞΙΑ", "ΤΕΛΙΚΗ ΑΞΙΑ"
            }
        ));
        scrollPane.setViewportViewView(table2);

JButton btnNewButton_2 = new
JButton("ΑΝΑΖΗΤΗΣΗ");
        btnNewButton_2.addActionListener(new
ActionListener() {

```

```
        public void
actionPerformed(ActionEvent e) {
            from= dateChooser.getDate();

            to= dateChooser_1.getDate();

            if(comboBox.getSelectedIndex()==0) {
                salePP();
            }
        }
    });
    btnNewButton_2.setBounds(621, 35, 94, 23);
    panelSales.add(btnNewButton_2);

    lblNewLabel_2 = new JLabel("");
    lblNewLabel_2.setBounds(10, 350, 113, 14);
    panelSales.add(lblNewLabel_2);

    label_4 = new JLabel("");
    label_4.setBounds(133, 350, 113, 14);
    panelSales.add(label_4);

    label_5 = new JLabel("");
    label_5.setBounds(243, 350, 113, 14);
    panelSales.add(label_5);

    label_6 = new JLabel("");
    label_6.setBounds(366, 350, 113, 14);
    panelSales.add(label_6);

    lblNewLabel_10 = new JLabel("");
    lblNewLabel_10.setBounds(499, 350, 94,
14);
    panelSales.add(lblNewLabel_10);

    label_7 = new JLabel("");
    label_7.setBounds(603, 350, 113, 14);
    panelSales.add(label_7);
```

```
}
```

```
public void panel3() {  
  
    panelprod = new JPanel();  
    panelprod.setBounds(145, 59, 725, 397);  
    contentPane.add(panelprod);  
    panelprod.setLayout(null);  
  
    JLabel label = new JLabel("Τυπος Αναφορας");  
    label.setFont(new Font("Tahoma", Font.PLAIN, 13));  
    label.setBounds(10, 11, 137, 14);  
    panelprod.add(label);  
  
    JLabel label_1 = new JLabel("Απο");  
    label_1.setFont(new Font("Tahoma", Font.PLAIN, 13));  
    label_1.setBounds(243, 11, 137, 14);  
    panelprod.add(label_1);  
  
    JLabel label_2 = new JLabel("Εως");  
    label_2.setFont(new Font("Tahoma", Font.PLAIN, 13));  
    label_2.setBounds(456, 11, 137, 14);  
    panelprod.add(label_2);  
  
    JComboBox comboBox = new JComboBox();  
    comboBox.setModel(new DefaultComboBoxModel(new String[]  
{"ΠΩΛΗΣΕΙΣ ΠΡΟΙΟΝΤΩΝ"})));  
    comboBox.setBounds(10, 36, 205, 20);  
    panelprod.add(comboBox);  
  
    Date date = new Date();  
    dateChooser = new JDateChooser();  
    dateChooser.setBounds(243, 36, 159, 20);  
    panelprod.add(dateChooser);  
    dateChooser.setDateFormatString("dd/MM/yyyy");  
    dateChooser.setDate(date);  
    dateChooser_1 = new JDateChooser();  
    dateChooser_1.setDateFormatString("dd/MM/yyyy");  
    dateChooser_1.setBounds(452, 36, 159, 20);  
    dateChooser_1.setDate(date);  
    panelprod.add(dateChooser_1);  
}
```

```

JScrollPane scrollPane = new JScrollPane();
scrollPane.setEnabled(false);
scrollPane.setBounds(10, 84, 705, 260);
panelprod.add(scrollPane);

table = new JTable();

table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
table.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "ID ΠΡΟΙΟΝΤΟΣ", "ΠΡΟΙΟΝ", "ΠΟΣΟΤΗΤΑ
ΠΩΛΗΣΕΩΝ", "ΜΕΣΗ ΤΙΜΗ", "ΣΥΝΟΛΟ ΠΩΛΗΣΕΩΝ"
    }
));
scrollPane.setViewportViewView(table);

JButton btnNewButton_2 = new
JButton("ΑΝΑΖΗΤΗΣΗ");
btnNewButton_2.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent e) {
        from= dateChooser.getDate();

        to= dateChooser_1.getDate();
        if(comboBox.getSelectedIndex()==0) {

            prodSS();

        }
    }
});
btnNewButton_2.setBounds(621, 35, 94, 23);
panelprod.add(btnNewButton_2);

JButton btnNewButton_3 = new
JButton("ΕΠΙΣΤΡΟΦΗ");
btnNewButton_3.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent e) {

        dispose();
    }
});
btnNewButton_3.setBounds(0, 401, 146, 55);
contentPane.add(btnNewButton_3);

```

```
}
```

```
public void contactsPaymentDue() {

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"ID ΠΕΛΑΤΗ",
"ΕΠΩΝΥΜΙΑ", "ΑΝΕΞΟΦΛΗΤΑ ΤΙΜΟΛΟΓΙΑ", "ΣΥΝ.ΠΟΣΟ"  });

    //InvoiceD in =new InvoiceD();
    ArrayList<InvoiceD> invoices =
InvoiceD.contactsBA(from, to);

    int t=0;
    int o=0;

    for(int y = 0; y < invoices.size(); y++) {

        System.out.println(invoices.size());
        Object[] row = new Object[4];
        ArrayList<ContactsD> contact =
ContactsD.FindContact(Integer.toString(invoices.get(y).getClientId
()));

        int g = 0;
        double h=0;

        for(int i = 0; i < invoices.size(); i++)
        {
            // ArrayList<ContactsD> contact =
ContactsD.FindContact(Integer.toString(invoices.get(i).getClientId
()));

            if(contact.get(0).getId()==invoices.get(i).getClientId())
            {

                g++;

                h+= invoices.get(i).getTotal();

                invoices.remove(i);

            }

        }

    }

}
```

```
        i=-1;
    }

}

    row[0] = contact.get(0).getId();
    row[1] = contact.get(0).getCname();
    row[2] = g;
    row[3] =h;
    t+=g;
    o+=h;

    model.addRow(row);
    y=-1;

}
table1.setModel(model);
lblNewLabel_1.setText(Integer.toString(t));
label_3.setText(Integer.toString(o));

}
```

```
public void contactsBiggerAmount() {

    System.out.println("te");

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setEnabled(true);
    scrollPane.setBounds(10, 84, 705, 260);
    panelcontacts.add(scrollPane);

    table1=new JTable();

    table1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION)
;
    table1.setModel(new DefaultTableModel(
        new Object[][] {
            },
        new String[] {
            "ID ΠΕΛΑΤΗ", "ΕΠΩΝΥΜΙΑ", "ΑΡ.ΤΙΜΟΛΟΓΙΩΝ",
"ΣΥΝ.ΠΟΣΟ"
        }
    ));
    scrollPane.setViewportViewView(table1);
    DefaultTableModel model = new DefaultTableModel();
```

```
        model.setColumnIdentifiers(new Object[]{"ID ΠΕΛΑΤΗ",
"ΕΠΩΝΥΜΙΑ", "ΑΡ.ΤΙΜΟΛΟΓΙΩΝ", "ΣΥΝ.ΠΟΣΟ"    });

        ArrayList<InvoiceD> invoices =
InvoiceD.contactsPaymDue(from, to);

        int t=0;
        int o=0;

        for(int y = 0; y < invoices.size(); y++) {

                Object[] row = new Object[4];
                ArrayList<ContactsD> contact =
ContactsD.FindContact(Integer.toString(invoices.get(y).getClientId
()));
                int g = 0;
                double h=0;

                for(int i = 0; i < invoices.size(); i++)
                {

if(contact.get(0).getId()==invoices.get(i).getClientId())
                {

                        g++;

                        h+= invoices.get(i).getTotal();

                        invoices.remove(i);

                        i=-1;

                }

                }

                row[0] = contact.get(0).getId();
                row[1] = contact.get(0).getCname();
                row[2] = g;
                row[3] =h;
                t+=g;
                o+=h;

```



```
        model.addRow(row);
            y=-1;
        }
        table1.setModel(model);
        TableRowSorter<TableModel> sorter = new
TableRowSorter<TableModel>(table1.getModel());
        table1.setRowSorter(sorter);
        ArrayList<RowSorter.SortKey> sortKeys = new
ArrayList<RowSorter.SortKey>();

        int columnIndexToSort = 3;
        sortKeys.add(new RowSorter.SortKey(columnIndexToSort,
SortOrder.DESCENDING));

        sorter.setSortKeys(sortKeys);
        sorter.sort();

        table1.setModel(model);

    }

    public void salePP() {

        DecimalFormat df = new DecimalFormat("#.##");
        DefaultTableModel model = new DefaultTableModel();
        model.setColumnIdentifiers(new Object[]{ "ID
TIM", "ΦΠΑ 6", "ΦΠΑ 13", "ΦΠΑ 24", "ΚΑΘΑΡΗ ΑΞΙΑ", "ΤΕΛΙΚΗ ΑΞΙΑ"});

        ArrayList<InvoiceD> invoices =
InvoiceD.contactsPaymDue(from, to);

        double fp6=0, fp13=0, fp24=0, fff=0, f1=0;

        for(int y = 0; y < invoices.size(); y++) {

            Object[] row = new Object[6];
            ArrayList<InvoiceD> items =
InvoiceD.InvoiceItems(Integer.toString(invoices.get(y).getId()));
```

```
double f6 = 0, f13=0, f24=0;
double ff=0, f=0;

for(int i = 0; i < items.size(); i++)
{
    double q,w;

    q=items.get(i).getpQuantity();
    w=items.get(i).getpPrice();

    if(items.get(i).getpTax()== 24 ) {
        f24+= Double.valueOf(df.format( (q*w)
* 24.0f / 100));

        }else if(items.get(i).getpTax()== 13) {
        f13+= Double.valueOf(df.format((q*w)
* 13.0f / 100));

        }else {
        f6+= Double.valueOf(df.format((q*w) *
6.0f / 100));

        }

        ff+= Double.valueOf(df.format(q*w));
//meriko

    }

    f= Double.valueOf(invoices.get(y).getTotal());

    row[0] = invoices.get(y).getId();
    row[1] = f6;
```

```
        row[2] = f13;
        row[3] =f24;
        row[4] =ff;
        row[5] =f;

        fp6+=f6;
        fp13+=f13;
        fp24+=f24;
        fff+=ff;
        f1+=f;

        model.addRow(row);

    }
    table2.setModel(model);

    lblNewLabel_2.setText("ΑΡ.ΤΙΜ: "+ invoices.size());
    label_4.setText("ΣΥΝ. :"+fp6);
    label_5.setText("ΣΥΝ. :"+fp13);
    label_6.setText("ΣΥΝ. :"+fp24);
    lblNewLabel_10.setText("ΣΥΝ. :"+fff);
    label_7.setText("ΣΥΝ. :"+f1);

}

public void prodSS() {

    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"ID
ΠΡΟΙΟΝΤΟΣ", "ΠΡΟΙΟΝ", "ΠΟΣΟΤΗΤΑ ΠΩΛΗΣΕΩΝ", "ΜΕΣΗ ΤΙΜΗ", "ΚΑΘΑΡΗ
ΑΞΙΑ", "ΣΥΝΟΛΟ ΤΕΛΙΚΟ" });

    int g = 0;
    double q=0,p=0;

    ArrayList<InvoiceD> items = InvoiceD.prodSS(from, to);

    for(int y = 0; y < items.size(); y++) {

        Object[] row = new Object[6];

        double mtimh=0;
        double it=0;
        double h=0;
        double tax=0;
```

```
for(int i = 1; i < items.size(); i++)
{
    if(items.get(0).getpId()==items.get(i).getpId())
    {

        g++;

        it+= items.get(i).getpQuantity();
        mtimh+=items.get(i).getpPrice();
        h++;

        items.remove(i);
        i=0;

    }

}

tax=items.get(0).getpTax();
it+=items.get(0).getpQuantity();
double mmtimh=(mtimh+items.get(0).getpPrice())/(h+1);

row[0] = items.get(0).getpId();
row[1]= items.get(0).getpName();
row[2] = it;
row[3] = mmtimh;
row[4] =it*mmtimh;
row[5] =df.format(( it*mmtimh* tax)/( 100 ) +
(it*mmtimh));
items.remove(0);
model.addRow(row);
y=-1;

}
table.setModel(model);

}
}//telos
```

## 5.7 Πακέτο settings

### 5.7.1 SettingsG.java

```
package settings;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import contacts.ContactsD;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import java.awt.Color;
import java.awt.Desktop;

import javax.swing.SwingConstants;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.JLayeredPane;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.awt.event.ActionEvent;

import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.sql.SQLException;

import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;

import com.mysql.jdbc.Blob;

import java.awt.Graphics2D;
```

```
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;
import java.awt.image.WritableRaster;
import java.io.IOException;
import javax.imageio.ImageIO;

public class SettingsG extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField_6;
    private JTextField textField_7;
    private JTextField textField_8;
    private JTextField textField_5;
    private JTextField textField_9;
    private JTextField textField_10;
    private JTextField textField_11;
    private JTextField textField_12;
    private JLabel lblimg;
    public String s;
    public File selectedFile;
    public FileInputStream is;

    public File newpic;
    public byte[] bytes =null;

    private static final int IMG WIDTH = 120;
    private static final int IMG HEIGHT = 120;
    JLabel label;
    ImageIcon photo;
    WritableRaster raster;
    DataBufferByte data;
    File image;
    private JPanel panel1;
    private JTextField textField_13;
    private JTextField textField_14;
    private JTextField textField_15;
    private JTextField textField_16;
```

```
public SettingsG() throws SQLException, IOException {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 847, 497);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    panel1 = new JPanel();
    panel1.setBounds(171, 0, 660, 458);
    contentPane.add(panel1);
    panel1.setLayout(null);
    JLabel lblNewLabel_1 = new JLabel("ΕΠΩΝΥΜΙΑ
ΕΠΙΧΕΙΡΗΣΗΣ");
    lblNewLabel_1.setBounds(10, 11, 165, 27);
    panel1.add(lblNewLabel_1);

    JLabel label_10 = new JLabel("ΔΙΑΚΡΙΤΟΣ ΤΙΤΛΟΣ");
    label_10.setBounds(323, 11, 165, 27);
    panel1.add(label_10);

    textField = new JTextField();
    textField.setBounds(10, 35, 281, 27);
    panel1.add(textField);
    textField.setColumns(10);

    textField_1 = new JTextField();
    textField_1.setColumns(10);
    textField_1.setBounds(323, 35, 294, 27);
    panel1.add(textField_1);

    JLabel label_1 = new JLabel("ΔΡΑΣΤΗΡΙΟΤΗΤΑ");
    label_1.setBounds(10, 73, 165, 27);
    panel1.add(label_1);

    textField_2 = new JTextField();
    textField_2.setColumns(10);
    textField_2.setBounds(10, 97, 281, 27);
    panel1.add(textField_2);

    JLabel label_2 = new JLabel("ΑΦΜ");
    label_2.setBounds(10, 135, 294, 27);
    panel1.add(label_2);

    textField_3 = new JTextField();
    textField_3.setColumns(10);
```

```
textField_3.setBounds(10, 159, 294, 27);
panel1.add(textField_3);

JLabel label_3 = new JLabel("ΔΟΥ");
label_3.setBounds(323, 135, 165, 27);
panel1.add(label_3);

textField_4 = new JTextField();
textField_4.setColumns(10);
textField_4.setBounds(323, 159, 281, 27);
panel1.add(textField_4);

textField_6 = new JTextField();
textField_6.setColumns(10);
textField_6.setBounds(10, 225, 281, 27);
panel1.add(textField_6);

JLabel label_5 = new JLabel("ΔΙΕΥΘΥΝΣΗ");
label_5.setBounds(10, 201, 165, 27);
panel1.add(label_5);

textField_7 = new JTextField();
textField_7.setColumns(10);
textField_7.setBounds(323, 225, 202, 27);
panel1.add(textField_7);

JLabel label_6 = new JLabel("ΠΕΡΙΟΧΗ");
label_6.setBounds(323, 201, 165, 27);
panel1.add(label_6);

textField_8 = new JTextField();
textField_8.setColumns(10);
textField_8.setBounds(535, 225, 105, 27);
panel1.add(textField_8);

JLabel label_7 = new JLabel("ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΑΣ");
label_7.setBounds(524, 197, 165, 27);
panel1.add(label_7);

JLabel label_4 = new JLabel("ΤΗΛΕΦΩΝΟ 1");
label_4.setBounds(10, 263, 165, 27);
panel1.add(label_4);

textField_5 = new JTextField();
textField_5.setColumns(10);
textField_5.setBounds(10, 287, 165, 27);
panel1.add(textField_5);

textField_9 = new JTextField();
```



```
textField_9.setColumns(10);
textField_9.setBounds(10, 354, 165, 27);
panel1.add(textField_9);

JLabel label_8 = new JLabel("ΤΗΛΕΦΩΝΟ 2");
label_8.setBounds(10, 325, 165, 27);
panel1.add(label_8);

textField_10 = new JTextField();
textField_10.setColumns(10);
textField_10.setBounds(185, 287, 165, 27);
panel1.add(textField_10);

JLabel label_9 = new JLabel("ΚΙΝΗΤΟ");
label_9.setBounds(185, 263, 165, 27);
panel1.add(label_9);

JLabel lblFax = new JLabel("Fax");
lblFax.setBounds(185, 325, 165, 27);
panel1.add(lblFax);

textField_11 = new JTextField();
textField_11.setColumns(10);
textField_11.setBounds(185, 354, 165, 27);
panel1.add(textField_11);

textField_12 = new JTextField();
textField_12.setColumns(10);
textField_12.setBounds(374, 287, 266, 27);
panel1.add(textField_12);

JLabel lblEmail = new JLabel("email");
lblEmail.setBounds(374, 263, 266, 27);
panel1.add(lblEmail);

JButton btnNewButton_1 = new JButton("ΑΝΑΖΗΤΗΣΗ");
btnNewButton_1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        JFileChooser file = new JFileChooser();
        file.setCurrentDirectory(new
File(System.getProperty("user.home")));
        //filter the files
        FileNameExtensionFilter filter = new
FileNameExtensionFilter("*.Images", "jpg", "gif", "png");
        file.addChoosableFileFilter(filter);
        int result = file.showOpenDialog(null);
        //if the user click on save in Jfilechooser
```

```
        if(result == JFileChooser.APPROVE_OPTION){

                selectedFile = file.getSelectedFile();
                String path =
selectedFile.getAbsolutePath();

                Image img1 =
Toolkit.getDefaultToolkit().createImage(path);

                lblimg.setIcon(new ImageIcon(new
ImageIcon(img1).getImage()).getScaledInstance(lblimg.getWidth(),
lblimg.getHeight(), Image.SCALE_DEFAULT));

        }

        else if(result ==
JFileChooser.CANCEL_OPTION){
                System.out.println("No File Select");
        }

    }

});
btnNewButton_1.setBounds(360, 356, 107, 23);
panel1.add(btnNewButton_1);

JButton button = new JButton("ΔΙΑΓΡΑΦΗ");
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        lblimg.setIcon(null);
        //selectedFile=null;

        try {
            FileOutputStream fos = null;

                byte[] t= {0};

                File file
=File.createTempFile("tempfile", ".jpg");

                fos = new
FileOutputStream(file);

                fos.write(t);
                fos.flush();
```

```
fos.close();

file.deleteOnExit();

selectedFile=file;
} catch (IOException ex) {
    // no application registered for
}

}
});
button.setBounds(360, 390, 107, 23);
panel1.add(button);

lblimg = new JLabel("");
lblimg.setBounds(477, 325, 178, 122);
panel1.add(lblimg);

JButton btnNewButton_2 = new JButton("ΕΠΙΣΤΡΟΦΗ");
btnNewButton_2.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e) {

        dispose();
    }
});
btnNewButton_2.setBounds(0, 420, 150, 38);
panel1.add(btnNewButton_2);

JButton button_2 = new JButton("ΕΦΑΡΜΟΓΗ");
button_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            changeMyCompany();

        } catch (FileNotFoundException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
```

```
        } finally {  
            dispose();  
        }  
    }  
});  
button_2.setBounds(154, 420, 150, 38);  
panel1.add(button_2);  
  
JPanel panel2 = new JPanel();  
panel2.setLayout(null);  
panel2.setBounds(171, 0, 660, 458);  
contentPane.add(panel2);  
panel2.setVisible(false);  
  
JLabel lblNewLabel = new JLabel("email :");  
lblNewLabel.setBounds(10, 60, 55, 23);  
panel2.add(lblNewLabel);  
  
textField_13 = new JTextField();  
textField_13.setBounds(75, 61, 208, 20);  
panel2.add(textField_13);  
textField_13.setColumns(10);  
  
JLabel label_11 = new JLabel("κωδικος :");  
label_11.setBounds(10, 94, 55, 23);  
panel2.add(label_11);  
  
textField_14 = new JTextField();  
textField_14.setColumns(10);  
textField_14.setBounds(75, 95, 208, 20);  
panel2.add(textField_14);  
  
JLabel lblEmailsmtpserver = new JLabel("emailSMTPserver  
:");  
lblEmailsmtpserver.setBounds(10, 127, 99, 23);  
panel2.add(lblEmailsmtpserver);  
  
textField_15 = new JTextField();  
textField_15.setColumns(10);  
textField_15.setBounds(119, 126, 208, 20);  
panel2.add(textField_15);  
  
JLabel label_12 = new JLabel("emailSMTPserver :");  
label_12.setBounds(10, 162, 99, 23);  
panel2.add(label_12);
```

```
textField_16 = new JTextField();
textField_16.setColumns(10);
textField_16.setBounds(119, 161, 208, 20);
panel2.add(textField_16);

JLabel lblEmail_1 = new JLabel("ΑΛΛΓΗ ΡΥΘΜΙΣΕΩΝ
EMAIL");
lblEmail_1.setBounds(10, 11, 173, 14);
panel2.add(lblEmail_1);

JButton btnNewButton_3 = new JButton("ΕΦΑΡΜΟΓΗ");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setEmail();
        dispose();
    }
});
btnNewButton_3.setBounds(401, 403, 121, 44);
panel2.add(btnNewButton_3);

JButton button_1 = new JButton("ΑΚΥΡΩΣΗ");
button_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
button_1.setBounds(529, 403, 121, 44);
panel2.add(button_1);

JButton btnNewButton = new JButton("ΣΤΟΙΧΕΙΑ
ΕΠΙΧΕΙΡΗΣΗΣ");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        panel2.setVisible(false);
        panel1.setVisible(true);
    }
});
btnNewButton.setBounds(0, 54, 169, 65);
contentPane.add(btnNewButton);

JButton btnEmail = new JButton("EMAIL");
btnEmail.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        panel1.setVisible(false);
        panel2.setVisible(true);
    }
});
```

```
        loadEmailDb();

    }
});
btnEmail.setBounds(0, 117, 169, 65);
contentPane.add(btnEmail);

fillmycompany();

}

public void fillmycompany() throws IOException{

    ContactsD mycompany=new ContactsD();
    mycompany.getMyCompany();

    textField.setText(mycompany.getCname());
    textField_1.setText(mycompany.getCtitle());
    textField_2.setText(mycompany.getOc());
    textField_3.setText(mycompany.getAfm());
    textField_4.setText(mycompany.getDoy());
    textField_6.setText(mycompany.getStreet());
    textField_7.setText(mycompany.getTown());
    textField_8.setText(mycompany.getZip());
    textField_5.setText(mycompany.getTel());
    textField_9.setText(mycompany.getTel2());
    textField_10.setText(mycompany.getMob());
    textField_11.setText(mycompany.getFax());
    textField_12.setText(mycompany.getEmail());
    if(mycompany.getLogo()!= null) {

        Image img =
Toolkit.getDefaultToolkit().createImage(mycompany.getLogo());

        try {

            FileOutputStream fos = null;
```

```

        File file
=File.createTempFile("tempfile", ".jpg");

        fos = new
FileOutputStream(file);

        fos.write(mycompany.getLogo());
        fos.flush();
        fos.close();

        file.deleteOnExit();

        selectedFile=file;

    } catch (IOException ex) {
        // no application registered for
PDFs
    }

        // BufferedImage bi = toBufferedImage(img); //
retrieve image

        //BufferedImage mapmodifiable = ImageIO.read(new
File("images/main.gif"));

        //FileInputStream fis=new FileInputStream();

        // ImageIO.read(mycompany.getLogo());
        // InputStream in = new
ByteArrayInputStream(mycompany.getLogo());
        //    BufferedImage bi = ImageIO.read(in);

        // selectedFile =File.createTempFile("tempfile",
".jpg");
        // ImageIO.write(bi, "jpg",selectedFile);

        lblimg.setIcon(new ImageIcon(new
ImageIcon(img).getImage().getScaledInstance(lblimg.getWidth(),
lblimg.getHeight(), Image.SCALE_DEFAULT)));
```

```
    }  
  
}  
  
public void changeMyCompany() throws FileNotFoundException  
{  
  
    ContactsD.updateMyCompany(textField,textField_1,  
textField_2, textField_3, textField_4, textField_6, textField_7,  
textField_8, textField_12, textField_5, textField_9,  
textField_10, textField_11, selectedFile);  
  
}  
  
public void loadEmailDb() {  
  
    ContactsD n=new ContactsD();  
    n.getSEmail();  
  
    textField_13.setText(n.getEm());  
    textField_14.setText(n.getPas());  
    textField_15.setText(n.getSser());  
    textField_16.setText(n.getSpor());  
  
    //ContactsD.updateEmail(textField_13,textField_14,textField_  
15,textField_16);  
}  
  
public void setEmail() {  
    ContactsD n=new ContactsD();  
  
    n.updateEmail(textField_13,textField_14,textField_15,textFie  
ld_16);
```



```
}  
  
public static BufferedImage toBufferedImage(Image img)  
{  
    if (img instanceof BufferedImage)  
    {  
        return (BufferedImage) img;  
    }  
  
    // Create a buffered image with transparency  
    BufferedImage bimage = new  
BufferedImage(img.getWidth(null), img.getHeight(null),  
BufferedImage.TYPE_INT_ARGB);  
  
    // Draw the image on to the buffered image  
    Graphics2D bGr = bimage.createGraphics();  
    bGr.drawImage(img, 0, 0, null);  
    bGr.dispose();  
  
    // Return the buffered image  
    return bimage;  
}  
  
}
```

### 5.7.2 Email.java

```
package settings;  
  
import java.io.*;  
import java.net.InetAddress;  
import java.util.Properties;  
import java.util.Date;  
  
import javax.activation.DataHandler;  
import javax.activation.DataSource;  
import javax.activation.FileDataSource;  
import javax.mail.*;  
  
import javax.mail.internet.*;  
  
import com.sun.mail.smtp.*;  
  
import contacts.ContactsD;
```

```
public class Email {

    String senderEmailID = null;
    String senderPassword = null;
    String emailSMTPserver = null;
    String emailServerPort = null;
    String receiverEmailID = null;
    static String emailSubject = "";
    static String emailBody = "";

    public Email(String receiverEmailID,String Subject,
String Body, byte[] bs){
        ContactsD n = new ContactsD();
        n.getSEmail();
        this.senderEmailID=n.getEm();
        this.senderPassword=n.getPas();
        this.emailSMTPserver=n.getSser();
        this.emailServerPort=n.getSpor();
        // Receiver Email Address
        this.receiverEmailID=receiverEmailID;
        // Subject
        Email.emailSubject=Subject;
        // Body
        Email.emailBody=Body;
        Properties props = new Properties();
        props.put("mail.smtp.user",senderEmailID);
        props.put("mail.smtp.host", emailSMTPserver);
        props.put("mail.smtp.port", emailServerPort);
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.socketFactory.port",
emailServerPort);
        props.put("mail.smtp.socketFactory.class",
"javax.net.ssl.SSLSocketFactory");
        props.put("mail.smtp.socketFactory.fallback", "false");
        SecurityManager security = System.getSecurityManager();
        try{
            Authenticator auth = new SMTPAuthenticator();
            Session session = Session.getInstance(props, auth);

            MimeMessage msg = new MimeMessage(session);
            msg.setText(emailBody);
            msg.setSubject(emailSubject);
            msg.setFrom(new InternetAddress(senderEmailID));
            msg.addRecipient(Message.RecipientType.TO,
new InternetAddress(receiverEmailID));
```

```
BodyPart messageBodyPart1 = new MimeBodyPart();
messageBodyPart1.setText(emailBody);

MimeBodyPart messageBodyPart2 = new MimeBodyPart();

messageBodyPart2.setDataHandler(new DataHandler(
    new DataSource() {

public String getContentType() {
    return "application/pdf";
}

public InputStream getInputStream() throws IOException {
    return new ByteArrayInputStream(bs);
}

public String getName() {
    return "myfile.pdf";
}

public OutputStream getOutputStream() throws IOException
{
    return null;
}}));

Multipart multipart = new MimeMultipart();
multipart.addBodyPart(messageBodyPart1);
multipart.addBodyPart(messageBodyPart2);

msg.setContent(multipart );

Transport.send(msg);
System.out.println("Message send Successfully:"); }

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

}
public class SMTPAuthenticator extends
javax.mail.Authenticator
```

```
        {  
            public PasswordAuthentication  
getPasswordAuthentication()  
        {  
            return new PasswordAuthentication(senderEmailID,  
senderPassword);  
        }  
    }  
  
}
```

### 5.7.3 GenerateInvoice.java

```
package settings;  
  
import java.io.ByteArrayOutputStream;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.net.MalformedURLException;  
import java.util.ArrayList;  
import com.itextpdf.layout.property.TextAlignment;  
import com.itextpdf.text.Document;  
import com.itextpdf.text.DocumentException;  
import com.itextpdf.text.Element;  
import com.itextpdf.text.Phrase;  
import com.itextpdf.text.Rectangle;  
import com.itextpdf.text.pdf.BaseFont;  
import com.itextpdf.text.pdf.PdfPCell;  
import com.itextpdf.text.pdf.PdfPTable;  
import com.itextpdf.text.pdf.PdfWriter;  
import com.itextpdf.text.Paragraph;  
import com.itextpdf.text.Font;  
import com.itextpdf.text.FontFactory;  
import com.itextpdf.text.Image;  
import com.itextpdf.text.PageSize;  
  
import contacts.ContactsD;  
import invoice.InvoiceD;  
  
public class GenerateInvoice {  
  
    private double f;  
    private double ff;
```

```
private double f24;
private double f13;
private double f6;
private ArrayList<InvoiceD> invoiceitems;
private ArrayList<ContactsD> contact;
public ArrayList<InvoiceD> invoice;
private Font font;
private ContactsD mycompany;

public static final String FONT =
"./src/fonts/arialuni.ttf";
public static final String DEST =
"C:\\Users\\mario\\Desktop\\back up\\m.pdf\\";

public GenerateInvoice(double f, double ff, double f24,
double f13, double f6, ArrayList<InvoiceD> invoiceitems,
ArrayList<ContactsD> contact, ArrayList<InvoiceD>
invoice) {

    this.f=f;
    this.ff=ff;
    this.f24=f24;
    this.f13=f13;
    this.f6=f6;
    this.invoiceitems=invoiceitems;
    this.contact=contact;
    this.invoice = invoice;

}

public void genpdf() {

try {
/*
    File file = new File(DEST);
    file.getParentFile().mkdirs();
    Document document = new Document();
    PdfWriter.getInstance(document, new
FileOutputStream(DEST));
    document.open();

*/
    Document document = new Document();
```

```
        ByteArrayOutputStream baos = new
ByteArrayOutputStream();
        PdfWriter.getInstance(document, baos);

        // PdfWriter.getInstance(document, new
FileOutputStream(DEST));
        document.open();

        font = FontFactory.getFont(FONT, BaseFont.IDENTITY_H ,
true);

        mycompany =new ContactsD();
        mycompany.getMyCompany();

        addTitle(document);
        addCompClient(document);
        addTable(document);
        addSum(document);

        document.close();
        InvoiceD.addPdfToDb(invoice.get(0).getId(),baos);

    } catch (IOException | DocumentException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public void addTitle(Document layoutDocument) throws
DocumentException, IOException
{
    Image img=Image.getInstance(mycompany.getLogo());
    img.scaleAbsolute(150f, 100f);

    PdfPTable table = new PdfPTable(3);

    table.setWidthPercentage(100);

    PdfPCell cell = new PdfPCell(img);
```

```
        cell.setFixedHeight(100);
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new Phrase(""));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        Paragraph par =new Paragraph("ΤΙΜΟΛΟΓΙΟ",font);
        par.add("\n\n");

        par.add("Αρ.Παραστατικού : " +
invoice.get(0).getId());
        par.add("\n\n");
        par.add("Ημερομ.Έκδοσης : " +
invoice.get(0).getInvoiceDate());
        cell = new PdfPCell(par);
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);
        layoutDocument.add(table);
        //table.addCell(getCell("Text in the middle",
PdfPCell.ALIGN_CENTER));
        //table.addCell(getCell("Text to the right",
PdfPCell.ALIGN_RIGHT));

    }

    public void addCompClient(Document layoutDocument) throws
DocumentException
    {

        PdfPCell cell;
        PdfPTable table = new PdfPTable(2);

        table.setWidthPercentage(100);

        cell = new PdfPCell(new Paragraph("ΑΠΟ",font));
        cell.setBorder(Rectangle.NO_BORDER);
```

```
        table.addCell(cell);

        cell = new PdfPCell(new
Paragraph("ΠΕΛΑΤΗΣ",font));
        cell.setBorder(Rectangle.NO_BORDER);

cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

        layoutDocument.add(table);

Paragraph n =new Paragraph(mycompany.getCtitle(),font);

layoutDocument.add(n);

table = new PdfPTable(2);

table.setWidthPercentage(100);

//row1
        cell = new PdfPCell(new
Paragraph(mycompany.getCname(),font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new
Paragraph(contact.get(0).getCname(),font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row2
        cell = new PdfPCell(new
Paragraph(mycompany.getOc(),font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new
Paragraph(contact.get(0).getOc(),font));
        cell.setBorder(Rectangle.NO_BORDER);
```



```
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row3
        cell = new PdfPCell(new Paragraph("ΑΦΜ: "+
mycompany.getAfm()+" ΔΟΥ: " + mycompany.getDoy(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new Paragraph("ΑΦΜ: "+
contact.get(0).getAfm() +" ΔΟΥ: " +
contact.get(0).getDoy(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row4
        cell = new PdfPCell(new
Paragraph(mycompany.getStreet(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new
Paragraph(contact.get(0).getStreet(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row6
        cell = new PdfPCell(new
Paragraph(mycompany.getTown()+" T.K: "+mycompany.getZip(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new
Paragraph(contact.get(0).getTown()+" T.K:
"+mycompany.getZip(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row7
        cell = new PdfPCell(new Paragraph("Τηλ1
"+mycompany.getTel() +" Τηλ2 "+mycompany.getTel2(), font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new Paragraph("", font));
        cell.setBorder(Rectangle.NO_BORDER);
```

```
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row8
        cell = new PdfPCell(new Paragraph("Κιν
"+mycompany.getMob() +" fax "+mycompany.getFax(),font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new Paragraph("  ",font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

//row9
        cell = new PdfPCell(new Paragraph("email
"+mycompany.getEmail(),font));
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        cell = new PdfPCell(new Paragraph("  ",font));
        cell.setBorder(Rectangle.NO_BORDER);
        cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
        table.addCell(cell);

        layoutDocument.add(table);

    }

    public void addTable(Document layoutDocument) throws
    DocumentException
    {

        Paragraph n =new Paragraph("\n");
        layoutDocument.add(n);

        PdfPTable table = new PdfPTable(new float[] { 3,1,1,1,1
});

        table.setWidthPercentage(100);

        // headers
        table.addCell(new Paragraph("ΠΡΟΙΟΝ",font));
```

```
        table.addCell(new Paragraph("ΠΟΣΟΤΗΤΑ", font));
        table.addCell(new Paragraph("ΤΙΜΗ", font));
        table.addCell(new Paragraph("ΦΠΑ", font));
        table.addCell(new Paragraph("ΣΥΝΟΛΟ", font));

        // items
        for(int i = 0; i < invoiceitems.size(); i++)
        {
            table.addCell(new
Paragraph(invoiceitems.get(i).getpName(), font));
            table.addCell(new
Paragraph(String.valueOf(invoiceitems.get(i).getpQuantity())));
            table.addCell(new
Paragraph(String.valueOf(invoiceitems.get(i).getpPrice())));
            table.addCell(new
Paragraph(String.valueOf(invoiceitems.get(i).getpTax())));
            table.addCell(new
Paragraph(String.valueOf(invoiceitems.get(i).getpQuantity() *
invoiceitems.get(i).getpPrice())));
        }

        layoutDocument.add(table);
    }
}
```

```
    public void addSum(Document layoutDocument) throws
DocumentException
    {

        Paragraph n =new Paragraph("\n");
        layoutDocument.add(n);

        PdfPTable table = new PdfPTable(new float[] {4,2 });

        table.setWidthPercentage(100);
        table.setHorizontalAlignment(Element.ALIGN_RIGHT);

        if( f24!= 0) {
            table.addCell(new Paragraph("ΦΠΑ
24", font));
            table.addCell(String.valueOf(f24));
        }
        if( f13!= 0) {
            table.addCell(new Paragraph("ΦΠΑ
13", font));
            table.addCell(String.valueOf(f13));
        }
    }
}
```

```
        if( f6!= 0) {
            table.addCell(new Paragraph("ΦΠΑ 6", font));
            table.addCell(String.valueOf(f6));
        }

        table.addCell(new Paragraph("ΣΥΝΟΛΟ ΦΠΑ", font));
        table.addCell(String.valueOf(f24+f13+f6));

        table.addCell(new Paragraph("ΣΥΝΟΛΟ
ΠΡΟΙΟΝΤΩΝ", font));
        table.addCell(String.valueOf(ff));

        table.addCell(new Paragraph("ΤΕΛΙΚΟ
ΣΥΝΟΛΟ", font));
        table.addCell(String.valueOf(f));

        layoutDocument.add(table);
    }
}
```

## ΚΕΦΑΛΑΙΟ 6

### ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι επιχειρήσεις είναι αναγκαίο να εξελίσσονται παράλληλα με την τεχνολογία και με τις αλλαγές που αυτή επιφέρει ώστε να μπορούν να καλύπτονται οι ανάγκες των πελατών και να παραμένουν ικανοποιημένοι με το ελάχιστο δυνατό κόστος.

Η εφαρμογή τιμολόγησης που υλοποιήθηκε έχει ορισμένα πλεονεκτήματα και είναι τα εξής :

- Γρήγορη και εύκολη αναζήτηση επαφών, τιμολογίων και προϊόντων.
- Εύκολη διαχείριση αποθήκης προϊόντων
- Εξοικονόμηση χρόνου για την έκδοση παραστατικού αφού τα στοιχεία (επαφές, προϊόντα) αποθηκεύονται και μπορούν να χρησιμοποιηθούν πολλές φορές.
- Έγκυρα στοιχεία πελάτη αφού μπορούν αυτόματα όλα τα στοιχεία του να αντληθούν από την υπηρεσία της ΓΓΠΣ
- Τροποποίηση στοιχείων επαφών και προϊόντων
- Μπορεί να χρησιμοποιηθεί από πολλούς χρήστες ταυτόχρονα και απομακρυσμένα αφού όλα τα δεδομένα της εφαρμογής βρίσκονται σε βάση δεδομένων
- Αποστολή παραστατικού με email απευθείας από την εφαρμογή, οπότε έτσι δεν χρειάζεται η εκτύπωση και η μεταφορά του τιμολογίου στον πελάτη
- Ευκολία στην εκμάθηση της

Η εφαρμογή έχει και ορισμένα μειονεκτήματα :

- Έλλειψη τρόπου διαγράψης δεδομένα όπως κάποια επαφή ή προϊόν
- Έλλειψη τρόπου ακύρωσης τιμολογίων
- Το τιμολόγιο που δημιουργείται σε μορφή PDF δεν είναι δομημένο οπότε δεν μπορεί να γίνει χρήση του για την εισαγωγή και επεξεργασία πληροφοριών του από άλλες λογιστικές εφαρμογές
- Έλλειψη τρόπου για εξόφληση του παραστατικού ηλεκτρονικά

Μελλοντικά η εφαρμογή μπορεί να αναπτυχθεί και να καλύψει τα παραπάνω μειονεκτήματα όπως:

- να γίνεται διαγραφή επαφών και προϊόντων αλλά με τέτοιο τρόπο ώστε να μην υπάρχει πρόβλημα με τα σχετιζόμενα παραστατικά που έχουν εκδοθεί
- αναφορά τιμολογίου ως ακυρωμένο ή διαγραφή του.
- δημιουργία δομημένου αρχείου PDF, δηλαδή να υπάρχουν μεταδεδομένα στο αρχείο που θα περιγράφουν ιδιότητες του παραστατικού όπως τα στοιχεία πελάτη και εμπόρου, τα προϊόντα, σύνολα κ.α. οπού ο χρήστης θα μπορεί να εξάγει και να κάνει επεξεργασία αυτά τα δεδομένα από άλλες εφαρμογές.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Excel applications for corporate finance. Boston: McGraw-Hill Irwin. - Adair, T. (2006).

[2] Η χρήση της Μηχανογραφημένης Λογιστικής στα λογιστικά γραφεία- Έρευνα στην πόλη της Θεσσαλονίκης, Πτυχιακή εργασία, Καβάλα: ΤΕΙ Καβάλας Παπαδοπούλου Βασιλική - Ούλιου Ηλιάνα (2011)

[3] Business documents. London: Mansell Publishing. - Armstrong, J. and Jones, S. (1987).

[5] MySQL. - DuBois, P. (n.d.).

[6] Pricing Decisions in the Euro Area. New York: Oxford University Press. - Fabiani, S., Loupias, C., Martins, F. και Sabbatini, R. (2007).

[7] Organizations. - Greenwald, H. (n.d.).

[8] Java programming. New York: Arcler Press. - Ishutin, O. (2015).

[9] Fiscal administration. - Mikesell, J.

[10] The management bible. Hoboken, N.J.: John Wiley & Sons. - Nelson, B. and Economy, P. (2005).

[11] Java Programming. Pearson India. - Pandey, H. (2012).

[12] The economic foundations of property rights. Cheltenham, UK: Edward Elgar. - Pejovich, S. (1997).

[13] ΗΛΕΚΤΡΟΝΙΚΗ ΤΙΜΟΛΟΓΗΣΗ, Πτυχιακή εργασία, ΤΕΙ Κρήτης. Available from: [http://nefeli.lib.teicrete.gr/browse/sdo/acfi/2014/RadosMichailTsampikos,Chatzipanagioto uEmmanouil/attached-document-1415191493-691937-3229/RadosMichail\\_ChatzipanagioutouEmmanouil2014.pdf](http://nefeli.lib.teicrete.gr/browse/sdo/acfi/2014/RadosMichailTsampikos,Chatzipanagioto uEmmanouil/attached-document-1415191493-691937-3229/RadosMichail_ChatzipanagioutouEmmanouil2014.pdf)

ΡΑΔΟΣ ΜΙΧΑΗΛ-ΤΣΑΜΠΙΚΟΣ - ΧΑΤΖΗΠΑΝΑΓΙΩΤΟΥ ΕΜΜΑΝΟΥΗΛ (2014)

[14] Ηλεκτρονική τιμολόγηση [online], Available from : [https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE\\_%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7#%CE%A3%CF%84%CF%8C%CF%87%CE%BF%CE%B9\\_%CF%84%CE%B7%CF%82\\_%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE%CF%82\\_%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7%CF%82\\_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9](https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE_%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7#%CE%A3%CF%84%CF%8C%CF%87%CE%BF%CE%B9_%CF%84%CE%B7%CF%82_%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE%CF%82_%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7%CF%82_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9):

Wikipedia.org

[15] Java programming. New York: McGraw-Hill Professional. - Sarang, P. (2012).

[16] Beginning MySQL. Indianapolis, IN: Wiley Pub. - Sheldon, R. and Moes, G. (2005).

[17] Financial intermediation in Europe. Boston: Kluwer Academic Publishers. - Spajić, L. (2002).



[18] The Java Programming Language. (2012). Prentice Hall.

[19] MySQL. Berkeley, CA: Peachpit Press. - Ullman, L. (2006).

[20] Assistive Technology. Amsterdam: BRILL. - Wendt, O. (2014).

[21] From Bureaucracy to Business Enterprise. - Whincop, M.

[22] ΔΙΟΝΑΤΟΣ, Γ. and ΔΙΟΝΑΤΟΣ, Γ. (2018). Η ηλεκτρονική τιμολόγηση εκτοξεύεται στα... σύννεφα. StartUpper Startup News.

<https://startupper.gr/%CE%B7->

[%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE-](https://startupper.gr/%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE-)

[%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7-](https://startupper.gr/%CF%84%CE%B9%CE%BC%CE%BF%CE%BB%CF%8C%CE%B3%CE%B7%CF%83%CE%B7-)

[%CE%B5%CE%BA%CF%84%CE%BF%CE%BE%CE%B5%CF%8D%CE%B5%CF%84%CE%B1/](https://startupper.gr/%CE%B5%CE%BA%CF%84%CE%BF%CE%BE%CE%B5%CF%8D%CE%B5%CF%84%CE%B1/)

..[23] Ηλεκτρονική τιμολόγηση Available from:

[http://www.logistis.gr/default.asp?pid=4&la=1&art\\_id=385&tb3=1](http://www.logistis.gr/default.asp?pid=4&la=1&art_id=385&tb3=1) , logistis.gr ,

Χαράλαμπος Τσουσης

[24] Η Διακίνηση και η Τιμολόγηση (Επισημάνσεις), Available from :

<http://foroline.gr/archives/3753> , Τασος Σακελλαροπουλος (2016)

[25] Ηλεκτρονική τιμολόγηση: Τέλος στα έντυπα παραστατικά.  
[https://www.huffingtonpost.gr/thanasis-mastoras/-\\_9280\\_b\\_13948046.html](https://www.huffingtonpost.gr/thanasis-mastoras/-_9280_b_13948046.html) -  
HuffPost Greece. (2018).

[26] Διαδικτυακή Υπηρεσία «Βασικά στοιχεία για νομικά πρόσωπα, νομικές οντότητες και φυσικά πρόσωπα με εισόδημα από επιχειρηματική δραστηριότητα»  
[http://www.gsis.gr/gsis/info/gsis\\_site/PublicIssue/wmsp/wmsp\\_pages/wmsp.html](http://www.gsis.gr/gsis/info/gsis_site/PublicIssue/wmsp/wmsp_pages/wmsp.html)