



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΑΠΟΘΗΚΗΣ ΚΑΤΑΣΤΗΜΑΤΟΣ ΜΕ ΧΡΗΣΗ JAVAFX

Του φοιτητή

Σιαμάτρα Ραφαήλ-Εμμανουήλ

Αρ. Μητρώου: 123929

Επιβλέπων καθηγητής

Κεραμόπουλος Ευκλείδης

Θεσσαλονίκη 2018

ΠΡΟΛΟΓΟΣ

Με την πάροδο του χρόνου, οι τεχνολογικές εξελίξεις πληθαίνουν όλο και περισσότερο. Αυτό έχει ως αποτέλεσμα, να μεγαλώνουν και οι απαιτήσεις των χρηστών σε κάθε τεχνολογικό τομέα. Για την ακρίβεια, θα μπορούσαμε να πούμε πως υπάρχει μια αλληλεξάρτηση μεταξύ αυτών των δύο εννοιών.

Για να γίνουν όλα αυτά υλοποιήσιμα όμως, θα πρέπει να υπάρχει και το κατάλληλο υπόβαθρο. Μένοντας στάσιμος στις τεχνικές ανάπτυξης τεχνολογικών προϊόντων, δεν μπορείς να ακολουθήσεις την εξελικτική τους πορεία.

Έτσι, και στο κομμάτι της ανάπτυξης εφαρμογών, από πλευράς Java, έχει αναπτυχθεί μία νέα πλατφόρμα, μέσω τη οποίας, ο προγραμματιστής έχει τη δυνατότητα να δημιουργήσει εφαρμογές οι οποίες ακολουθούν τις νέες τεχνολογικές τάσεις όπως η αφή, η διεισδυτικότητα του διαδικτύου σε όλες σχεδόν τις ηλεκτρονικές συσκευές, ενώ παράλληλα κάνει ευκολότερο και αποδοτικότερο το έργο του προγραμματιστή.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία στοχεύει στην ανάδειξη της JavaFX, μία τεχνολογία την οποία η Java έχει ορίσει ως διάδοχο της Swing στη δημιουργία διεπαφών χρήστη. Ωστόσο, η Swing συνεχίζει να παραμένει στο προσκήνιο καθώς, είναι απαραίτητος κάποιος χρόνος προσαρμογής για την πλήρη μετάβαση στη νέα πλατφόρμα λογισμικού.

Με μία σύγκριση των δύο τεχνολογιών, καταλαβαίνει κανείς την υπερίσχυση της JavaFX έναντι της Swing. CSS, Scene Builder, FXML, Component's Properties και Media Engine, είναι μερικοί από τους όρους που κάνουν τη JavaFX να υπερισχύει του “αντιπάλου” της.

Μέσω της εφαρμογής που αναπτύχθηκε για τις ανάγκες της παρούσας πτυχιακής εργασίας, μπορεί κανείς να πάρει μία πρώτη γεύση, της νέας αυτής τεχνολογίας.

ABSTRACT

This thesis aims at highlighting of JavaFX, a technology that Java has defined as Swing's successor in creating of user interfaces. However, Swing continues to be in the forefront as it is a necessary time to adapt to a complete transition to the new software platform.

By comparing these two technologies, everyone understands the predominance of JavaFX over Swing. CSS, Scene Builder, FXML, Component's Properties and Media Engine, are some of the terms that make JavaFX override its "rival".

Through the application developed for the purposes of this thesis, anyone can get a first taste of this new technology.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Κεραμόπουλο Ευκλείδη, ο οποίος αρχικά με εμπιστεύτηκε, επιτρέποντάς μου να αναλάβω μία πτυχιακή το θέμα της οποίας το πρότεινα εγώ ο ίδιος, και ο οποίος παρά το φόρτο της εργασίας του, επικοινωνούσε άμεσα μαζί μου, όποτε εγώ το χρειαζόμουν. Πάνω απ' όλα όμως, θα ήθελα να τον ευχαριστήσω για τον ανθρώπινο χαρακτήρα που επιδεικνύει και σαν καθηγητής αλλά και σαν Άνθρωπος.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους οικείους μου οι οποίοι με στήριξαν κατά το διάστημα της δημιουργίας της παρούσας πτυχιακής εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	3
ΠΕΡΙΛΗΨΗ.....	4
ABSTRACT.....	5
ΕΥΧΑΡΙΣΤΙΕΣ.....	6
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ.....	9
ΕΙΣΑΓΩΓΗ.....	10
ΚΕΦΑΛΑΙΟ 1: Η πλατφόρμα λογισμικού JavaFX.....	11
ΕΙΣΑΓΩΓΗ.....	11
1.1 Η εξέλιξη της JavaFX.....	12
1.2 Χαρακτηριστικά της JavaFX.....	13
1.3 Ιεραρχία της JavaFX.....	14
1.4 JavaFX CSS.....	16
1.4.1 Τρόποι απόδοσης style στα γραφικά στοιχεία.....	17
1.4.2 Σύνδεση εφαρμογής και αρχείων CSS.....	17
1.4.3 Το CSS στην εφαρμογή.....	18
1.5 JavaFX Scene Builder.....	19
1.5.1 Χαρακτηριστικά του Scene Builder.....	19
1.6 Σύνοψη κεφαλαίου.....	21
ΚΕΦΑΛΑΙΟ 2: Διαφορές μεταξύ JavaFX και Swing.....	22
ΕΙΣΑΓΩΓΗ.....	22
2.1 Διαφοροποιήσεις μεταξύ των δύο τεχνολογιών.....	23
2.2 Βελτιώσεις πάνω στις ήδη υπάρχουσες δυνατότητες.....	24
2.3 Προσθήκη νέων τεχνολογιών.....	24
2.4 Σύνοψη κεφαλαίου.....	26
ΚΕΦΑΛΑΙΟ 3: Χρησιμοποιούμενες βιβλιοθήκες κώδικα για την ανάπτυξη της εφαρμογής.....	27
ΕΙΣΑΓΩΓΗ.....	27
3.1 Η βιβλιοθήκη iText.....	28
3.1.1 Το iText στην εφαρμογή.....	29
3.1.1.1 Δημιουργία κεφαλίδας/υποσέλιδου.....	30
3.1.1.2 Δημιουργία πίνακα.....	34
3.1.1.3 Δημιουργία κεφαλίδων εγγράφου.....	36
3.1.1.4 Εισαγωγή των στοιχείων στο έγγραφο.....	36
3.2 Η βιβλιοθήκη γραφικών JFoenix.....	37
3.2.1 Λήψη και εγκατάσταση του JFoenix.....	38
3.2.2 Το JFoenix στην εφαρμογή.....	39

3.3 Η βάση δεδομένων και η επικοινωνία με αυτή.....	40
3.4 Σύνοψη κεφαλαίου.....	41
ΚΕΦΑΛΑΙΟ 4: Η εφαρμογή “Warehouse Management”.....	42
ΕΙΣΑΓΩΓΗ.....	42
4.1 Περιγραφή της Διεπαφής Χρήστη.....	43
4.1.1 Κύρια διεπαφή – Προεπισκόπηση της αποθήκης.....	43
4.1.2 Κύρια διεπαφή – Δημιουργία προϊόντος.....	44
4.1.3 Κύρια διεπαφή – Ημερήσιες πωλήσεις.....	45
4.1.4 Δευτερεύουσα διεπαφή – Σχετικά με την εφαρμογή.....	46
4.2 Περιγραφή του κώδικα της εφαρμογής.....	47
4.2.1 Βασικά βήματα για τη λειτουργία μίας JavaFX εφαρμογής.....	48
4.2.2 Διαχείριση (events) γεγονότων.....	49
4.2.3 Δημιουργία νέων Windows Borders.....	51
4.2.3.1 Drag & Drop παραθύρου.....	52
4.2.4 Διαχείριση φωτογραφιών προϊόντων.....	53
4.2.4.1 Αποθήκευση φωτογραφίας στη βάση δεδομένων.....	53
4.2.4.2 Ανάκτηση και απεικόνιση φωτογραφίας.....	53
4.2.5 Περιγραφή μεθόδων εφαρμογής.....	54
4.2.5.1 Αναζήτηση προϊόντος.....	54
4.2.5.2 Ανανέωση φόρμας προϊόντος.....	54
4.2.5.3 Εισαγωγή ακροατών στην εισαγωγή δεδομένων.....	54
4.2.5.4 JavaFX Dialogs.....	55
4.2.5.5 Προγραμματιστική μορφοποίηση του στυλ.....	58
4.3 Εναλλακτικοί τρόποι υλοποίησης της εφαρμογής.....	59
4.3.1 Ολοκλήρωση πώλησης στην περίπτωση μηδενικού αποθέματος.....	59
4.3.2 Δυνατότητα επεξεργασίας των στοιχείων κάθε προϊόντος από το αριστερό μενού της κύριας σελίδας.....	60
4.3.3 Δημιουργία προσαρμοσμένου Tooltip.....	60
4.4 Προτάσεις ανάπτυξης της εφαρμογής.....	60
4.4.1 Διαχείριση προμηθευτών.....	61
4.4.2 Αυτόματη δημιουργία παραγγελίας.....	61
4.4.3 Καταχώρηση πελατών.....	61
4.5 Σύνοψη κεφαλαίου.....	62
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	63
ΑΝΑΦΟΡΕΣ.....	64
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	65

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1: Αρχιτεκτονική της JavaFX	14
Εικόνα 2: Το Scene Builder ενσωματωμένο στο NetBeans.....	20
Εικόνα 3: Η εξέλιξη του iText.....	28
Εικόνα 4: Το Material Design της Google	37
Εικόνα 5: Η επίσημη ιστοσελίδα του JFoenix	38
Εικόνα 6: Η κεντρική σελίδα της εφαρμογής.....	43
Εικόνα 7: Φόρμα δημιουργίας νέου προϊόντος.....	45
Εικόνα 8: Διαχείριση των ημερήσιων πωλήσεων	46
Εικόνα 9: Παράθυρο περιγραφής της εφαρμογής	47

ΕΙΣΑΓΩΓΗ

Μία εφαρμογή αυτού του αντικειμένου, μπορεί να περιέχει μεγάλη πληθώρα δυνατοτήτων, όπως διαχείριση προϊόντων, προμηθευτών και πελατών, καταχώρηση τιμολογίων, διαχείριση παραγγελιών και πολλές άλλες. Ωστόσο, στην παρούσα πτυχιακή εργασία, απώτερος στόχος δεν ήταν τόσο η ανάπτυξη ενός λογισμικού που να ανταποκρίνεται σε κάθε ανάγκη που έχει να κάνει με τη διαχείριση μίας αποθήκης, όσο η παρουσίαση της τεχνολογίας JavaFX με την οποία δημιουργήθηκε η εφαρμογή. Γι' αυτό το λόγο, η εφαρμογή δεν εστιάζει σε κάτι περισσότερο από τη διαχείριση των προϊόντων της αποθήκης. Ωστόσο, ο στόχος της πτυχιακής, ο οποίος είναι η παρουσίαση την νέας τεχνολογίας της Java για τη δημιουργία διεπαφών χρήστη, της JavaFX, επιτυγχάνεται μέσω της εφαρμογής, η οποία φέρει τα χαρακτηριστικότερα συστατικά στοιχεία της JavaFX.

Αρχικά, γίνεται περιγραφή της JavaFX, παρουσιάζονται τα χαρακτηριστικά και οι κύριες ιδιότητες της, καθώς και τα συστατικά στοιχεία τα οποία αποτελούν την ιεραρχία της τεχνολογίας αυτής.

Στο δεύτερο κεφάλαιο, γίνεται μία προσπάθεια σύγκρισης της JavaFX με τον προκάτοχο της, τη Swing, αναφέροντας τα κυρία χαρακτηριστικά της κάθε τεχνολογίας τα οποία τις χαρακτηρίζουν.

Στο τρίτο κεφάλαιο, παρουσιάζονται δύο βιβλιοθήκες μέσω των οποίων υλοποιούνται, μία βασική δυνατότητα της εφαρμογής, να δημιουργεί PDF αρχεία, και μία ιδιότητα της εφαρμογής, η οποία είναι η εισαγωγή κάποιων συστατικών στοιχείων της διεπαφής, τα οποία βασίζονται πάνω στο Material Design, δημιουργήματα της Google.

Τέλος, γίνεται μία αναλυτική περιγραφή της εφαρμογής, παρουσιάζοντας κάποιες από τις βασικότερες ιδιότητες της, καθώς και προτάσεις βελτίωσης και περαιτέρω ανάπτυξης της.

ΚΕΦΑΛΑΙΟ 1

Η πλατφόρμα λογισμικού JavaFX

ΕΙΣΑΓΩΓΗ

Το Δεκέμβριο του 2008, η τότε Sun Microsystems κυκλοφόρησε την έκδοση 1.0.2 της JavaFX. Ήταν η νέα πλατφόρμα λογισμικού που ήρθε για να δώσει έναν διαφορετικό τρόπο προσέγγισης στην ανάπτυξη εφαρμογών. Πλέον ως παράγωγο της Java, παρέχεται από την Oracle Corporation.

Η JavaFX χρησιμοποιώντας ήδη υπάρχουσες τεχνολογίες όπως τη γλώσσα σήμανσης XML μέσω της FXML και με τη βοήθεια του Scene Builder, του CSS (Cascading Style Sheet) κ.α. δίνει στον προγραμματιστή τη δυνατότητα να διαχειρίζεται καλύτερα το υλικό του, με πιο ξεκάθαρο τρόπο, καθώς και τη δυνατότητα να δώσει άλλη διάσταση στη γραφική υπόσταση της διεπαφής.

1.1 Η εξέλιξη της JavaFX

Αρχικά η Sun Microsystems είχε κυκλοφορήσει την JavaFX στην πρώτη της έκδοση ως μία γλώσσα δέσμης ενεργειών (script language) που ως σκοπό είχε τη δημιουργία RIAs (Rich Internet Application). Συνεπώς, ξέφευγε από την αντικειμενοστραφή γλώσσα προγραμματισμού της, δηλαδή την Java και κατά συνέπεια χρειαζόταν η εκμάθηση μίας νέας γλώσσας προκειμένου κάποιος να αναπτύξει μία εφαρμογή σε JavaFX. Από τη στιγμή όμως που η Java πέρασε στα χέρια της Oracle, αυτή με τη σειρά της αποφάσισε να ενσωματώσει τη JavaFX μέσα στη Java. Έτσι, από την έκδοση 2.0, αποτελεί βιβλιοθήκη της Java[1].

Από το 2014 βρίσκεται στο Java SE Runtime Environment (JRE) και στο Java Development Kit (JDK). Κατά συνέπεια, υποστηρίζεται από όλα τα μεγάλα desktop λειτουργικά συστήματα (Windows, Mac OS και Linux). Μάλιστα, από την έκδοση JavaFX 8, υποστηρίζεται και από πλατφόρμες αρχιτεκτονικής ARM, δηλαδή και από λειτουργικά κινητών συσκευών.

Τα γνωστότερα IDE (Integrated Development Environment) όπως τα Netbeans, Eclipse και IntelliJ IDEA παρέχουν αρκετές δυνατότητες και λειτουργίες προς διευκόλυνση του προγραμματιστή, όπως μικρών ολοκληρωμένων εφαρμογών σε JavaFX που βοηθούν στην καλύτερη κατανόηση της, ενσωμάτωση του Scene Builder σε αυτά για πιο άμεση διαχείριση της γραφικής διεπαφής, επεξεργαστή XML αρχείων καθώς και οπτικής αποσφαλμάτωσης της διεπαφής χωρίς να γίνεται έλεγχος του κώδικα.

1.2 Χαρακτηριστικά της JavaFX

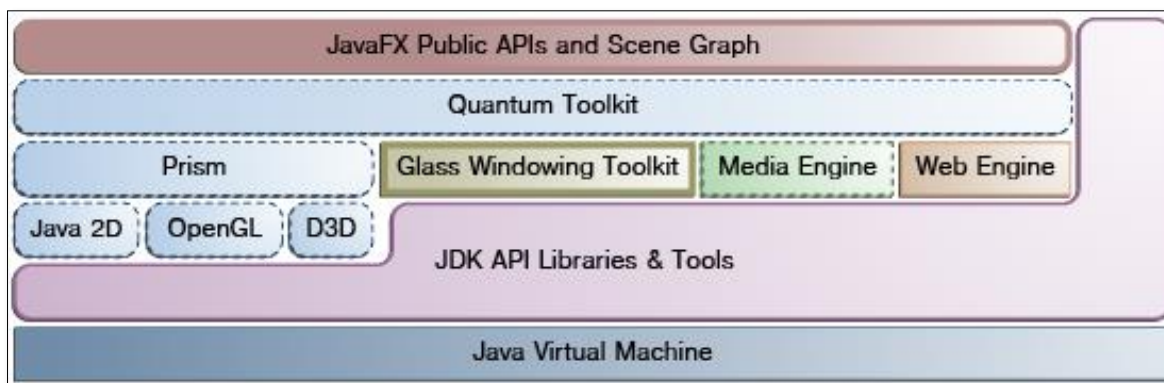
Παρακάτω παρουσιάζονται τα περισσότερα από τα χαρακτηριστικά της JavaFX. Πιο συγκεκριμένα όμως, γίνεται εστίαση στη έκδοση 8:

- **Java APIs:** Η JavaFX είναι μία βιβλιοθήκη Java η οποία αποτελείται από κλάσεις και interfaces τα οποία είναι γραμμένα σε κώδικα Java. Τα APIs έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να είναι μία φιλική εναλλακτική λύση στις γλώσσες της Java εικονικής μηχανής (JVM languages), όπως η JRuby και η Scala.
- **FXML και Scene Builder:** Η FXML είναι μία γλώσσα δηλωτικής σήμανσης βασισμένη στην XML που χρησιμοποιείται για τη δημιουργία διεπαφής χρήστη για μία εφαρμογή JavaFX. Ο σχεδιαστής μπορεί να γράψει κώδικα σε FXML ή να χρησιμοποιήσει το οπτικό εργαλείο διαμόρφωσης, JavaFX Scene Builder

για να σχεδιάσει διαδραστικά τη γραφική διεπαφή χρήστη (GUI). Το Scene Builder παράγει τον κώδικα FXML όπου μπορεί να μεταφερθεί σε ένα IDE όπου εκεί ο προγραμματιστής μπορεί να προσθέσει την προγραμματιστική λογική.

- **WebView:** Ένα χρήσιμο στοιχείο της JavaFX είναι το WebView. Χρησιμοποιεί την τεχνολογία WebKitHTML για να κάνει δυνατή την ενσωμάτωση ιστοσελίδων στην JavaFX εφαρμογή. Επίσης, υπάρχει μία αμφίδρομη επικοινωνία μεταξύ της javascript που τρέχει μέσα στο WebView και των Java APIs. Στην έκδοση 8 της JavaFX, υποστηρίζονται περισσότερα χαρακτηριστικά της HTML5 συμπεριλαμβανομένων και των Web Sockets, Web Workers και Web Fonts καθώς και δυνατότητες εκτύπωσης.
- **Διαλειτουργικότητα της Swing:** Προϋπάρχουσες εφαρμογές που έχουν δημιουργηθεί πάνω στη Swing μπορούν να ενημερωθούν με χαρακτηριστικά της JavaFX όπως η αναπαραγωγή γραφικών πολυμέσων και η ενσωμάτωση περιεχομένου ιστού. Από την έκδοση 8, η JavaFX δίνει και την αντίστροφη δυνατότητα δηλαδή, μέσα σε μία JavaFX εφαρμογή να ενσωματώσεις περιεχόμενο της Swing. Αυτό μπορεί να επιτευχθεί με την κλάση SwingNode[2].
- **Στοιχεία ελέγχου και CSS:** Η JavaFX παρέχει όλα τα σημαντικά στοιχεία ελέγχου ενός User Interface (UI) που απαιτούνται για την ανάπτυξη μιας πλήρους εφαρμογής. Τα συστατικά αυτά μπορούν να μορφοποιηθούν με τις τυπικές τεχνολογίες ιστού, όπως το CSS δίνοντας την ιδανικότερη εικόνα που θα πρέπει να έχουν για την εκάστοτε εφαρμογή. Από την έκδοση 8 έχουν προστεθεί τα στοιχεία DatePicker και TableView.
- **Προκαθορισμένο θέμα μορφοποίησης:** Η JavaFX μέχρι την έκδοση 2.2 είχε ως προκαθορισμένο θέμα μορφοποίησης των στοιχείων ελέγχου της το Caspian. Από την έκδοση 8 το Modena αντικατέστησε τον προκάτοχο του.
- **Χαρακτηριστικά 3D γραφικών:** Το API της JavaFX παρέχει μία βιβλιοθήκη για τη δημιουργία και χρήση γραφικών τριών διαστάσεων.
- **Υποστήριξη πολλαπλής αφής:** Παρέχει υποστήριξη για λειτουργίες πολλαπλής αφής, με την προϋπόθεση να το υποστηρίζει η πλατφόρμα στην οποία λειτουργεί η εφαρμογή.
- **Υποστήριξη του Hi-DPI:** Από την έκδοση 8, η JavaFX υποστηρίζει την τεχνολογία Hi-DPI. Ο όρος αυτός ανήκει στην Apple και αναφέρεται στις περιπτώσεις στις οποίες μία συσκευή διαθέτει οθόνη τύπου Retina όπου κάθε γραφικό περιβάλλον χρήστη, διπλασιάζεται σε πλάτος και ύψος για να αντισταθμιστεί για τα μικρότερα εικονοστοιχεία (pixels)[3].

1.3 Ιεραρχία της JavaFX



Εικόνα 1: Αρχιτεκτονική της JavaFX

Παρακάτω γίνεται περιγραφή στα πεδία της αρχιτεκτονικής τα οποία ανήκουν αποκλειστικά στη JavaFX.

Scene Graph:

Αποτελεί μέρος του κορυφαίου στρώματος στην αρχιτεκτονική της JavaFX. Είναι το σημείο εκκίνησης για την κατασκευή μίας εφαρμογής. Πρόκειται για ένα ιεραρχικό δέντρο κόμβων που αντιπροσωπεύει όλα τα οπτικά στοιχεία της διεπαφής χρήστη. Μπορεί να χειριστεί τις εισόδους και να οπτικοποιηθεί.

Ένα στοιχείο σε ένα γράφημα σκηνης ονομάζεται κόμβος. Κάθε κόμβος έχει αναγνωριστικό, κατηγορία style και οριοθέτηση ορίων. Με εξαίρεση τον κόμβο root ενός γραφήματος σκηνης, κάθε κόμβος έχει έναν μοναδικό γονέα και κανένα, ένα ή περισσότερα παιδιά.

JavaFX Public APIs:

Στην κορυφή της αρχιτεκτονικής της JavaFX βρίσκεται επίσης ένα σύνολο API τα οποία υποστηρίζουν την ανάπτυξη πλούσιων εφαρμογών πελάτη (Rich Client Application) και έχουν τα εξής χαρακτηριστικά:

- Επιτρέπει τη χρήση ισχυρών λειτουργιών της Java, όπως των generics, των annotations, της πολυεπεξεργασίας(multithreading) και των expressions Lamda.

- Διευκολύνει τους προγραμματιστές ιστού να χρησιμοποιούν την JavaFX από άλλες δυναμικές γλώσσες που βασίζονται στη JVM, όπως η Groovy και η JavaScript.
- Επιτρέπει στους προγραμματιστές Java να χρησιμοποιούν άλλες γλώσσες συστήματος, σε περίπτωση μεγάλων ή και σύνθετων JavaFX εφαρμογών.
- Τα API της JavaFX και το προγραμματιστικό μοντέλο αποτελούν συνέχεια της αρχικής έκδοσης της JavaFX. Τα περισσότερα από τα API της JavaFX έχουν μεταφερθεί απευθείας στην Java. Ορισμένα API, όπως το Layout και το Media, μαζί με πολλές άλλες λεπτομέρειες, βελτιώθηκαν και απλοποιήθηκαν με βάση τα σχόλια χρηστών της έκδοσης JavaFX 1.x. Το JavaFX βασίζεται περισσότερο στα πρότυπα ιστού, όπως το CSS για τα στοιχεία ελέγχου του στυλ και το ARIA για τις προδιαγραφές προσβασιμότητας.

Γραφικό σύστημα:

Τα Quantum Toolkit, Prism, Java 2D, OpenGL και D3D αποτελούν το γραφικό σύστημα της JavaFX και βρίσκονται στο δεύτερο στρώμα της αρχιτεκτονικής. Το γραφικό σύστημα υποστηρίζει δισδιάστατα και τρισδιάστατα γραφήματα σκηνής και υποβοηθά στην οπτικοποίηση τους όταν το υλικό του συστήματος είναι ανεπαρκές για την επαρκή επιτάχυνση της εκτέλεσης του υλικού.

Glass Windowing Toolkit:

Το Glass Windowing Toolkit βρίσκεται στο χαμηλότερο επίπεδο στη στοίβα του γραφικού συστήματος της JavaFX. Η κύρια ευθύνη του είναι να παρέχει εγγενείς λειτουργικές υπηρεσίες, όπως η διαχείριση των παραθύρων, των χρονιστών(timers) και των επιφανειών(surfaces). Χρησιμεύει ως το επίπεδο το οποίο είναι εξαρτώμενο από την πλατφόρμα και συνδέει την πλατφόρμα της JavaFX με το λειτουργικό σύστημα.

Το Glass Toolkit είναι επίσης υπεύθυνο για τη διαχείριση της ουράς συμβάντων. Σε αντίθεση με το Abstract Window Toolkit(AWT) το οποίο διαχειρίζεται τη δική του ουρά συμβάντων, το Glass Toolkit χρησιμοποιεί τη λειτουργικότητα της ουράς συμβάντων του λειτουργικού συστήματος για να προγραμματίσει τη χρήση του νήματος(thread). Επίσης, σε αντίθεση με το AWT, το Glass Toolkit τρέχει στο ίδιο νήμα με την JavaFX, ενώ στο AWT, το εγγενές μισό του AWT τρέχει σε ένα νήμα και το επίπεδο της Java τρέχει σε άλλο νήμα. Αυτό μπορεί να δημιουργήσει πολλά προβλήματα.

Media Engine:

Το JavaFX υποστηρίζει οπτικά και ακουστικά μέσα. Παρέχει υποστήριξη για αρχεία ήχου τύπου MP3, AIFF και WAV και αρχεία βίντεο FLV. Η λειτουργία πολυμέσων JavaFX παρέχεται ως τρία ξεχωριστά στοιχεία: το αντικείμενο Media αντιπροσωπεύει ένα αρχείο πολυμέσων, το MediaPlayer αναπαράγει ένα αρχείο

πολυμέσων και το MediaView είναι ένας κόμβος που εμφανίζει τα μέσα. Το Media Engine έχει σχεδιαστεί με γνώμονα την απόδοση και τη σταθερότητα και παρέχει σταθερή συμπεριφορά σε όλες τις πλατφόρμες.

Web Engine:

Το Web Component είναι ένα από τα στοιχεία μίας διεπαφής χρήστη JavaFX, βασισμένο στο Webkit, το οποίο παρέχει τη δυνατότητα προβολής περιεχομένου Web και την περιήγηση σε αυτό μέσω του δικού του API.

Το Webkit πάνω στο οποίο βασίζεται το Web Component, είναι ένας μηχανισμός προγράμματος περιήγησης ιστού, ανοικτού κώδικα, που υποστηρίζει τεχνολογίες όπως, HTML5, CSS, JavaScript, DOM και SVG.

Το Web Component παρέχει τις παρακάτω λειτουργίες:

- Προβολή HTML περιεχομένου από τοπική ή απομακρυσμένη διεύθυνση URL.
- Υποστήριξη διατήρησης ιστορικού, δίνοντας τη δυνατότητα να υπάρχει πλοήγηση προς τα πίσω ή εμπρός.
- Ανανέωση του περιεχομένου.
- Εισαγωγή εφέ στο Web Component.
- Επεξεργασία του HTML περιεχομένου.
- Εκτέλεση εντολών JavaScript.
- Χειρισμό συμβάντων (event handling).

Το Web Engine παρέχει τη βασική δυνατότητα περιήγησης σε ιστοσελίδες. Έτσι, το WebView παίρνει ένα αντικείμενο WebEngine, ενσωματώνει το HTML περιεχόμενο στο scene της εφαρμογής και παρέχει πεδία και μεθόδους για την εφαρμογή εφέ και μετασχηματισμών.

1.4 JavaFX CSS

Η JavaFX δίνει τη δυνατότητα στον προγραμματιστή να οπτικοποιήσει όπως αυτός θέλει τη διεπαφή της εφαρμογής του. Αυτό επιτυγχάνεται μέσω του JavaFX CSS.

Το JavaFX CSS βασίζεται στην έκδοση 2.1 του CSS της W3C με κάποιες προσθήκες από την έκδοση 3. Έτσι, μεγάλο πλεονέκτημα πάνω στο κομμάτι του CSS έχουν οι προγραμματιστές ιστού οι οποίοι είναι ήδη εξοικειωμένοι με το CSS για την HTML.

Το JavaFX CSS εφαρμόζεται στα στοιχεία της JavaFX με τρόπο παρόμοιο με αυτόν που εφαρμόζεται το CSS στα στοιχεία του HTML DOM. Εφαρμόζεται πρώτα στο γονικό κόμβο και έπειτα στα παιδιά του.

Ένας κόμβος δέχεται το style αφού έχει προστεθεί στο scene της εφαρμογής. Από κει και πέρα, το style εφαρμόζεται ξανά όταν υπάρξει αλλαγή στην ψευδό-κλάση του κόμβου, στο style-class, στο id ή στο ενσωματωμένο στον κώδικα style.

Το JavaFX CSS δεν υποστηρίζει τις ιδιότητες διάταξης του CSS όπως οι float, position, overflow και η width. Ωστόσο, οι ιδιότητες padding και margin υποστηρίζονται σε κάποια αντικείμενα της JavaFX. Οι υπόλοιπες ιδιότητες της διάταξης προγραμματίζονται μέσω του κώδικα της εφαρμογής.

1.4.1 Τρόποι απόδοσης style στα γραφικά στοιχεία

Κάθε κόμβος έχει μία μεταβλητή styleClass τύπου λίστας που περιέχει κάποια Strings. Είναι αντίστοιχο με το χαρακτηριστικό class="..." που υπάρχει στα στοιχεία της HTML. Έτσι, μπορούμε να ορίσουμε διάφορες κλάσεις style τις οποίες μπορούν να υιοθετήσουν οι κόμβοι. Για να δημιουργήσουμε μία styleClass αρκεί στο αρχείο CSS, η δέσμη εντολών που ανήκουν στην ίδια κλάση, να έχει ως κεφαλίδα το ".styleClass".

Ένας κόμβος περιέχει επίσης μία μεταβλητή id τύπου String. Είναι αντίστοιχο με το χαρακτηριστικό id="..." που υπάρχει στα στοιχεία της HTML. Έχουν παρόμοια λειτουργία με τα styleClass, μόνο που το κάθε id αντιπροσωπεύει τα χαρακτηριστικά style ενός συγκεκριμένου κόμβου. Για να δημιουργήσουμε ένα id αρκεί στο αρχείο CSS, η δέσμη εντολών που ανήκουν στο ίδιο αναγνωριστικό, να έχει ως κεφαλίδα το "#id".

Πέραν των επεμβατικών τρόπων για να δώσουμε style στα γραφικά στοιχεία της JavaFX, κάθε ένα από αυτά ανήκει σε κάποια κλάση (διαφορετική από την styleClass) της JavaFX που τα προσδίδει ένα προκαθορισμένο style.

Ένας άλλος τρόπος για να δώσουμε style σε μία εφαρμογή εκτός των συμπεριλαμβανομένων CSS αρχείων στην εφαρμογή, είναι μέσω του προγραμματιστικού κώδικα Java της εφαρμογής. Αυτό γίνεται μέσω της μεθόδου .setStyle. Η μέθοδος αυτή έχει τη υψηλότερη προτεραιότητα στην ανάθεση style στους κόμβους.

1.4.2 Σύνδεση εφαρμογής και αρχείων CSS

Για να συνδέσουμε ένα αρχείο CSS με την εφαρμογή θα πρέπει να δηλώσουμε με την κατάλληλη μέθοδο, τη διαδρομή του αρχείου. Η διαδρομή μπορεί να είναι απόλυτη ("C:\Users\rafail\style.css") ή σχετική ("style.css"). Αν ισχύει η δεύτερη περίπτωση, τότε το θέμα είναι να ορίσουμε μία θέση ως

προκαθορισμένη. Αυτό γίνεται με τη βασική διεύθυνση URL του classLoader από την κύρια κλάση της εφαρμογής. Ο ευκολότερος τρόπος για να δηλώσουμε τη διαδρομή του αρχείου είναι μέσω της μεθόδου `.getStyleSheet()` της κλάσης `Scene`: `scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());`

1.4.3 Το CSS στην εφαρμογή

Στο σημείο αυτό γίνεται αναφορά στη διεπαφή της εφαρμογής, παρουσιάζοντας κάποια από τα στοιχεία της διεπαφής της, τα οποία έχουν μορφοποιηθεί μέσω του JavaFX CSS.

Όπως έχει ήδη αναφερθεί, ο τρόπος με τον οποίο μορφοποιείται μία διεπαφή JavaFX μέσω του CSS είναι σχεδόν ίδιος με αυτόν που μορφοποιείται μία ιστοσελίδα HTML. Η Oracle παρέχει το δικό της API για το CSS της κάθε έκδοσης της JavaFX.

```
.combo-box-popup .list-view .list-cell{
    -fx-text-fill:#2c7db4;
}
```

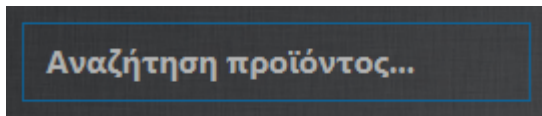
Μία διαφορά μεταξύ των CSS των JavaFX και W3C είναι ότι, οι ιδιότητες των στοιχείων, φέρουν στην αρχή του ονόματός τους, το πρόθεμα “-fx-”.

```
#searchField {
    -fx-background-color: transparent;
    -fx-background-radius:0;
    -fx-border-width:1px;
    -fx-border-color:#0e6297;
    -fx-text-fill:#c5d0bc;
    -fx-font-weight:bold;
    -fx-font-size:16;
}
```

Στο παραπάνω παράδειγμα, ορίζεται ότι το στοιχείο με `id = “searchField”` θα έχει τις παρακάτω ιδιότητες:

- Το background θα είναι διαφανές.
- Θα είναι απόλυτα τετραγωνισμένο.
- Το πάχος του περιγράμματος του θα είναι 1px.
- Το χρώμα του περιγράμματος θα είναι #0e6297.
- Το χρώμα του περιεχομένου θα είναι #c5d0bc.
- Το περιεχόμενο του θα είναι σε **bold**.
- Η γραμματοσειρά του περιεχομένου του θα είναι 16.

Το αποτέλεσμα του παραπάνω κώδικα είναι το παρακάτω:



Να σημειωθεί επίσης ότι το CSS της JavaFX δεν υποστηρίζει τα pseudo-Elements τα οποία υπάρχουν σε αυτό της W3C ωστόσο υποστηρίζονται τα pseudo-Classes κάνοντας εντονότερη την αίσθηση αλληλεπίδρασης του χρήστη με την εφαρμογή.

1.5 JavaFX Scene Builder

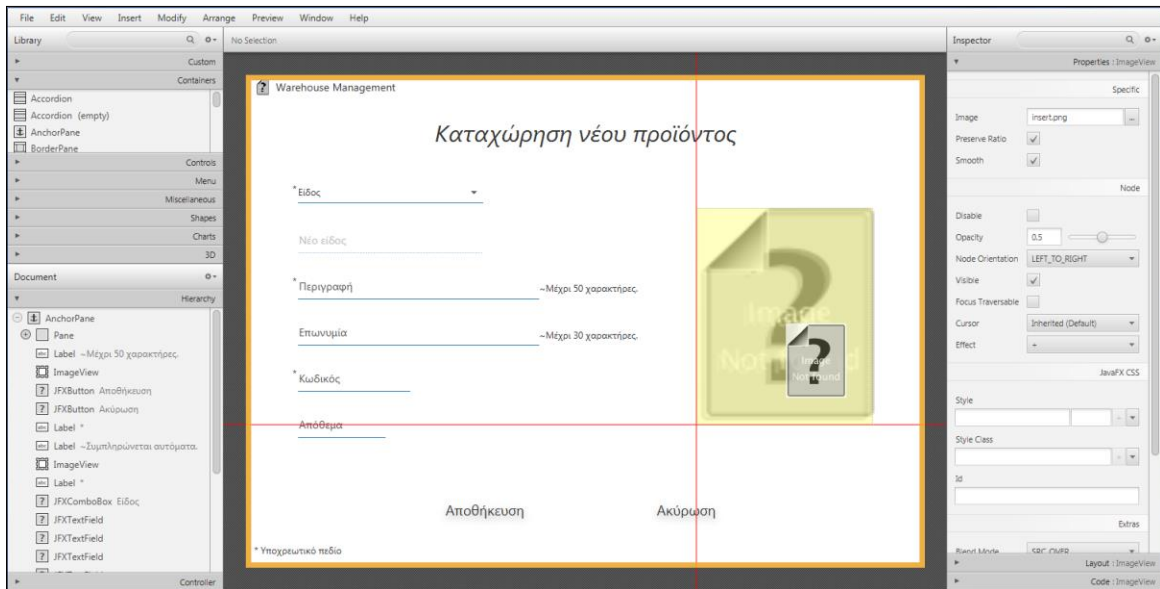
Το Scene Builder είναι το εργαλείο το οποίο κάνει τη δημιουργία ενός JavaFX έργου αρκετά πιο εύκολη, γρήγορη και αποτελεσματική, παρέχοντας τη δυνατότητα να μην χρειάζεται απαραίτητα η γραφή κώδικα για τη δημιουργία της διεπαφής χρήστη (UI). Αυτό επιτυγχάνεται μέσω της λειτουργίας drag and drop των συστατικών στοιχείων μέσα στο scene της εφαρμογής. Παράλληλα, δημιουργείται αυτόματα ο κώδικας FXML της διεπαφής χρήστη.

Με αυτό τον τρόπο, διαχωρίζονται μεταξύ τους ως ένα μέτρο, η δημιουργία της διεπαφής χρήστη με την προγραμματιστική λογική της εφαρμογής. Ωστόσο, μέσω της απλής αλλά αρκετά βοηθητικής διεπαφής του Scene Builder, παρέχεται επίσης η δυνατότητα σε άτομα που δεν έχουν άμεση σχέση με τον προγραμματισμό, να δημιουργήσουν κάποια πρότυπα διαδραστικών και λειτουργικών εφαρμογών τα οποία εμπεριέχουν ως ένα μέτρο, τη λογική της εφαρμογής.

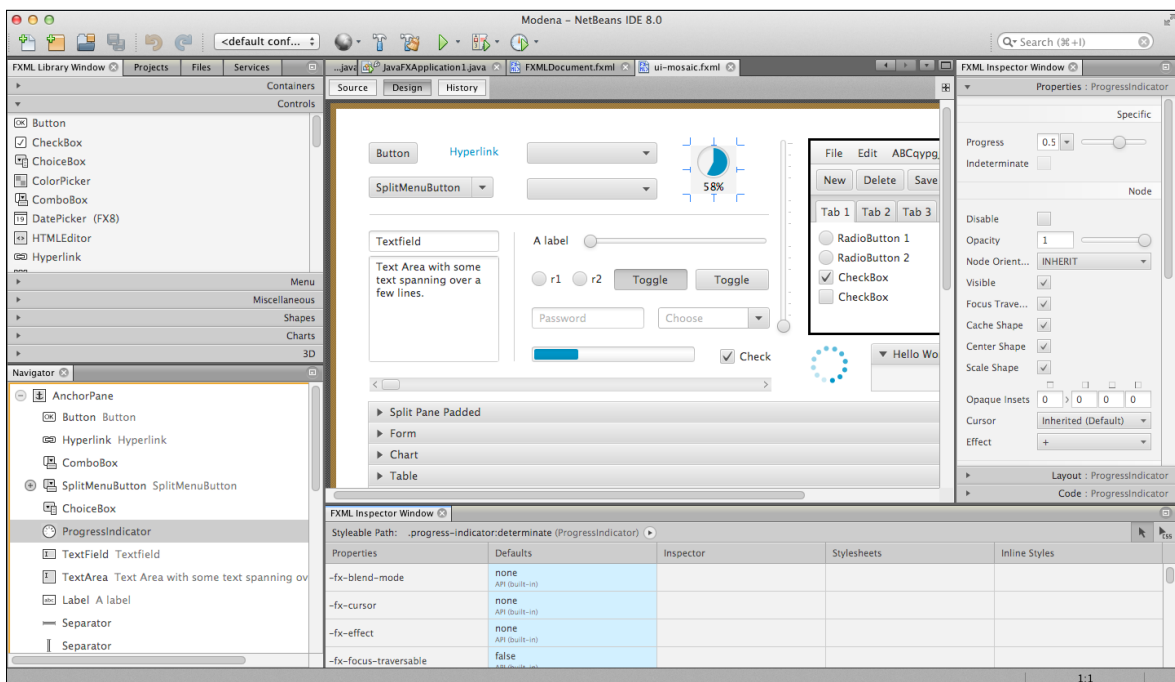
1.5.1 Χαρακτηριστικά του Scene Builder

- Όπως προαναφέρθηκε, το Scene Builder δίνει τη δυνατότητα ο σχεδιαστής να προσθέτει τα components στη διεπαφή χρήστη μέσω του drag & drop. Αυτό βοηθά στη μείωση του χρόνου δημιουργίας της διεπαφής αποφεύγοντας τη συγγραφή κώδικα.

Πτυχιακή εργασία του φοιτητή Σιαμάτρα Ραφαήλ-Εμμανουήλ



- Η ενσωμάτωση του Scene Builder με τα γνωστότερα IDE όπως το NetBeans, παρέχει τη βέλτιστη ροή εργασιών ανάπτυξης.



Εικόνα 2: Το Scene Builder ενσωματωμένο στο NetBeans

- Ο αυτόνομος κώδικας που δημιουργείται αυτόματα κατά τη διάρκεια δημιουργίας της διεπαφής χρήστη, αποθηκεύεται σε ξεχωριστό αρχείο από την προγραμματιστική λογική και το αρχείο CSS.

DailySales	18/11/2017 3:55 μμ	Αρχείο FXML	8 KB
DailySalesController	17/11/2017 7:22 μμ	Αρχείο JAVA	19 KB
dailySalesCSS	16/11/2017 3:03 πμ	CSS Document	3 KB

- Η λειτουργία ζωντανής επεξεργασίας και προεπισκόπησης επιτρέπει τη λειτουργία γρήγορης απεικόνισης του αποτελέσματος κατά τη σχεδίαση και επεξεργασία της διεπαφής χωρίς να χρειάζεται να μεταγλωττίζεται κάθε φορά ο κώδικας. Επίσης άμεσο, είναι το αποτέλεσμα σε περίπτωση που οριστεί κάποιο CSS αρχείο.
- Μέσω της διεπαφής του Scene Builder, ο προγραμματιστής μπορεί να ορίσει μεθόδους στους χειριστές γεγονότων, να ορίσει τιμές στις ιδιότητες των components καθώς και να ορίσει την κλάση controller, από την οποία, η διεπαφή θα παίρνει την προγραμματιστική λογική.

1.6 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό έγινε μία γενικευμένη και συνοπτική περιγραφή της JavaFX, με την παρουσίαση των κυριότερων χαρακτηριστικών της, και την περιγραφή των επιπέδων που δομούν την ιεραρχία της. Επίσης, παρουσιάστηκαν δύο πολύ βασικές προσθήκες στην δημιουργία διεπαφών χρήστη, το CSS, προϋπάρχον στον κλάδο της ανάπτυξης διαδικτυακών εφαρμογών και το Scene Builder, ένα άκρως πολύτιμο εργαλείο στα χέρια του προγραμματιστή με το οποίο η διαδικασία δημιουργίας της διεπαφής γίνεται αρκετά πιο εύκολη, εξοικονομώντας παράλληλα πολύτιμο χρόνο.

ΚΕΦΑΛΑΙΟ 2

Διαφορές μεταξύ JavaFX και Swing

ΕΙΣΑΓΩΓΗ

Για πολλά χρόνια η Swing αποτελούσε ένα εξαιρετικά αποτελεσματικό εργαλείο για την κατασκευή γραφικών διεπαφών χρήστη (GUI) και την προσθήκη αλληλεπίδρασης σε εφαρμογές Java. Ωστόσο, ο καιρός περνάει και αυξάνονται οι απαιτήσεις των χρηστών και δημιουργούνται νέες λειτουργίες πάνω στα GUI. Έτσι, η Swing έχει φτάσει σε ένα σημείο να μη μπορεί να αποδώσει όλες αυτές τις δυνατότητες. Παρόλο που η Oracle έχει σταματήσει να υποστηρίζει και να εξελίσσει τη Swing, δεν την έχει εγκαταλείψει πλήρως, δίνοντας ακόμα τη δυνατότητα ανάπτυξης GUI σε αυτή. Μάλιστα υπάρχει η δυνατότητα ένταξης κώδικα Swing σε JavaFX και το αντίστροφο. Βέβαια η Oracle έχει καταστήσει σαφές ότι το μέλλον στην ανάπτυξη GUI είναι η JavaFX.

Στο κεφάλαιο αυτό παρουσιάζονται οι κυριότερες διαφορές μεταξύ των δύο τεχνολογιών. Οι διαφορές αυτές ομαδοποιούνται σε τρεις κατηγορίες: Διαφοροποιήσεις μεταξύ των δύο τεχνολογιών, βελτιώσεις που έγιναν στην JavaFX πάνω σε χαρακτηριστικά τα οποία παρέχονται και από τις δύο τεχνολογίες και νέες προσθήκες που έγιναν στη JavaFX.

2.1 Διαφοροποιήσεις μεταξύ των δύο τεχνολογιών

Σε μία JavaFX εφαρμογή, όλα βρίσκονται μέσα σε ένα Stage

Στην Swing, η κλάση στην οποία εμπεριέχονται όλα τα στοιχεία της διεπαφής χρήστη ονομάζεται frame και ορίζεται από την κλάση JFrame. Το frame είναι ουσιαστικά ένα κενό παράθυρο μέσα στο οποίο προστίθεται ένα panel το οποίο λειτουργεί ως “δοχείο” μέσα στο οποίο προστίθενται τα στοιχεία της διεπαφής. Μία εφαρμογή Swing ουσιαστικά είναι μία κλάση η οποία κληρονομεί την κλάση JFrame.

Η JavaFX από την άλλη, χρησιμοποιεί τη μεταφορά ενός θεάτρου για να μοντελοποιήσει τα containers (“δοχεία”) της εφαρμογής. Ένα stage (θέατρο) αναπαριστά το “δοχείο” του ψηλότερου επιπέδου το οποίο συνήθως είναι ένα παράθυρο ή και ολόκληρη η επιφάνεια εργασίας του λειτουργικού. Τα επιμέρους στοιχεία της διεπαφής εμπεριέχονται σε ένα scene (σκηνή). Μία εφαρμογή μπορεί να έχει παραπάνω από μία scene αλλά μόνο μία μπορεί να εμφανίζεται στο stage ανά πάσα στιγμή.

Μία σκηνή περιέχει ένα γράφημα σκηνής (scene graph), το οποίο είναι και η σημαντικότερη έννοια στην JavaFX. Το γράφημα σκηνής είναι μία συλλογή όλων των στοιχείων όπως τα groups, layouts, controls και shapes τα οποία συγκροτούν μία διεπαφή χρήστη. Όλα αυτά τα στοιχεία ορίζονται από την κλάση Node. Η κλάση αυτή παρέχει πολλά χαρακτηριστικά στο χρήστη, με τα οποία μπορεί να διαχειριστεί με τον καλύτερο δυνατό τρόπο όλα αυτά τα στοιχεία.

Στην JavaFX, διαχειριστές του layout είναι όλα τα nodes

Στη Swing, ένα layout συσχετίζεται με ένα JPanel. Το πρώτο στην ιεραρχία JPanel δίνει το βασικό layout για ένα frame. Αν χρειαστεί να τοποθετηθεί ένα πρόσθετο layout μέσα στο πρώτο ιεραρχικά layout, θα πρέπει να δημιουργηθεί ένα επιπλέον JPanel, να οριστεί σε αυτό το δεύτερο layout και έπειτα να προστεθεί και αυτό με τη σειρά του στην ιεραρχία των layouts.

Στη JavaFX, τα layouts είναι υποκλάσεις της κλάσης Node όπως τα controls και τα shapes. Έτσι μέσα σε ένα layout μπορεί να υπάρχει οποιοδήποτε είδους στοιχείο της κλάσης Node όπως τα controls και τα shapes αλλά και κάποιο άλλο layout. Αυτή η λογική είναι πρακτικά πιο εύχρηστη απ’ ότι αυτή της Swing με τη συσχέτιση των layouts και των panels.

2.2 Βελτιώσεις πάνω στις ήδη υπάρχουσες δυνατότητες

Βελτιωμένος χειρισμός γεγονότων (Event Handling)

Τόσο η JavaFX όσο και η Swing χρησιμοποιούν το χειρισμό γεγονότων για να ανταποκρίνονται στην αλληλεπίδραση μεταξύ συστήματος και χρήστη. Ωστόσο, στην JavaFX ο χειρισμός γεγονότων έχει μελετηθεί και αναπτυχθεί εκτενέστερα για την καλύτερη δυνατή αξιοποίηση τους. Ο μεγαλύτερος όμως λόγος, βελτίωσης του event handling είναι η εξάρτηση του από τις ιδιότητες (properties).

Περισσότερα controls

Στη JavaFX έχει γίνει προσπάθεια να υπάρξει μία μεγαλύτερη γκάμα controls απ' ό τι στη Swing. Και οι δύο διαθέτουν όλα τα βασικά στοιχεία ελέγχου (buttons, check boxes, combo boxes κτλ). Ωστόσο, στη JavaFX έχουν προστεθεί κάποια ενδιαφέροντα controls που η Swing δεν έχει, όπως τα Titled Pane, τα Accordions κ.α.

Ευκολότερη χρήση κινουμένων σχεδίων (Animations)

Η Swing παρέχει τη δυνατότητα δημιουργίας animations σε μία εφαρμογή. Ωστόσο, δεν υπάρχει άμεση υποστήριξη για αυτό. Θα πρέπει να οριστούν χειροκίνητα οι timers και γενικώς, η λογική δημιουργίας μίας κινούμενης εικόνας θα πρέπει να αναπτυχθεί από τον προγραμματιστή.

Από τη άλλη, η JavaFX παρέχει ενσωματωμένη υποστήριξη για την ανάπτυξη κινουμένων σχεδίων η οποία μπορεί να εφαρμοστεί σε κάθε node του Scene Graph. Υπάρχει μία πληθώρα ενσωματωμένων κλάσεων για να ορίσετε κοινά animations όπως οι αποχρώσεις, οι περιστροφές ή οι διαδρομές κίνησης. Επίσης, υπάρχει η δυνατότητα προσαρμοσμένου animation μέσω των κλάσεων KeyFrame και Timeline.

2.3 Προσθήκη νέων τεχνολογιών

Χρήση της FXML

Η Swing δεν προσέφερε κάποια δηλωτική προσέγγιση για την ανάπτυξη διεπαφών χρήστη. Αυτό το έκανε η JavaFX μέσω της FXML. Μία γλώσσα σήμανσης βασισμένη στην XML, η οποία επιτρέπει τη δημιουργία της διεπαφής χρήστη ξεχωριστά από την προγραμματιστική λογική της εφαρμογής. Αυτό βοηθά

τον προγραμματιστή να αναπτύσσει και να συντηρεί ευκολότερα τη διεπαφή χρήστη χωρίς να χρειάζεται να επεμβαίνει απαραίτητα στις κώδικα της εφαρμογής.

JavaFX Scene Builder

Το εργαλείο σχεδίασης Scene Builder παρέχεται από την JavaFX και χρησιμοποιείται για την ανάπτυξη του layout των εφαρμογών. Με τη δυνατότητα drag and drop ο προγραμματιστής μπορεί να εισάγει όποιο component θέλει από τη συλλογή της JavaFX στο παράθυρο επεξεργασίας προκειμένου να δώσει σάρκα και οστά στην εφαρμογή του. Ταυτόχρονα, ο Scene Builder παράγει τον κώδικα FXML της διεπαφής σε ένα αρχείο με κατάληξη “.fxml” και αυτό με τη σειρά του ενσωματώνεται στην υπόλοιπη εφαρμογή. Το αρχείο αυτό μπορεί να επεξεργαστεί κάποιος από οποιονδήποτε text editor όπως φυσικά και από τα IDE όπως τα NetBeans και Eclipse.

JavaFX CSS

Ένα από τα σημαντικότερα χαρακτηριστικά της JavaFX είναι ότι ο προγραμματιστής μπορεί να μορφοποιήσει όπως αυτός θέλει την εφαρμογή μέσω του CSS. Καθετί στη διεπαφή μπορεί να μορφοποιηθεί ξεχωριστά από όλα τα υπόλοιπα στοιχεία. Επίσης, αναλόγως τη περίπτωση (περιπτώσεις χρήσης, δεδομένων καταστάσεων κ.α.) τα διάφορα components της εφαρμογής μπορούν να μορφοποιηθούν αντίστοιχα. Έτσι, δίνεται η δυνατότητα στον προγραμματιστή να έχει το πάνω χέρι στην οπτικοποίηση της εφαρμογής σε κάθε χρονική στιγμή.

Properties

Η JavaFX υποστηρίζει την έννοια των ιδιοτήτων σε όλες τις κλάσεις. Μία ιδιότητα είναι ουσιαστικά μία μεταβλητή που μπορεί να παρακολουθείται. Μέσω ενός listener σε μία ιδιότητα, ο προγραμματιστής έχει τη δυνατότητα να τρέξει κάποιο συγκεκριμένο κομμάτι κώδικα όταν η τιμή της ιδιότητας αλλάζει. Επίσης, υπάρχει η δυνατότητα, δύο ιδιότητες να αλληλοεξαρτώνται, δηλαδή όταν αλλάζει η τιμή της μίας, να αλλάζει και η τιμή της δεύτερης.

Υποστήριξη Media

Με την υποστήριξη πολυμέσων που παρέχει η JavaFX, ο προγραμματιστής της εφαρμογής μπορεί να προσθέσει πολυμεσικό περιεχόμενο σε αυτή όπως, αναπαραγωγή ήχου και εικόνας.

HTML περιεχόμενο

Από τη στιγμή που το διαδίκτυο απέκτησε την δύναμη που έχει σήμερα, υπήρχε και υπάρχει ακόμα η ανάγκη ενσωμάτωσης του σε σχεδόν όλη την

τεχνολογική κοινωνία. Έτσι, από την εποχή της Swing ακόμα, υπήρχε η ανάγκη ενσωμάτωσης HTML περιεχομένου στην ανάπτυξη των εφαρμογών.

Η JavaFX ικανοποίησε αυτή την ανάγκη, παρέχοντας ένας στοιχείο διεπαφής χρήστη το οποίο έχει την πλήρη ικανότητα προβολής ιστού και λειτουργικότητα περιήγησης.

Special Effects

Η JavaFX μέσω του πακέτου `javafx.scene.effect` παρέχει τη δυνατότητα να εφαρμοστούν κάποια ειδικά εφέ σε κάθε κόμβο του scene graph. Έτσι, ο προγραμματιστής μπορεί να εφαρμόσει σκιές, αντανάκλασεις, blurs (θολά σημεία) και άλλα ενδιαφέροντα χαρακτηριστικά, μεταμορφώνοντας την εμφάνιση της διεπαφής χρήστη τόσο για πρακτικούς όσο και για αισθητικούς λόγους.

Υποστήριξη συσκευών αφής

Με την JavaFX, πλέον υπάρχει η δυνατότητα αλληλεπίδρασης του χρήστη με την διεπαφή μέσω της αφής, κάτι το οποίο η Swing δεν παρείχε. Έτσι, μέσω του απλοποιημένου μηχανισμού διαχείρισης συμβάντων, ο προγραμματιστής μπορεί να διαχειριστεί κοινές χειρονομίες αφής, όπως η κύλιση, η ταλάντωση, η περιστροφή και η μεγέθυνση, εισάγοντας τον κατάλληλο ακροατή (listener) στο αντίστοιχο γεγονός αφής (touch event).

2.4 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό έγινε σύγκριση μεταξύ των δύο τεχνολογιών ανάπτυξης διεπαφών χρήστη, τονίζοντας τόσο τα χαρακτηριστικά της μίας όσο και της άλλης. Ο καθένας καταλαβαίνει πως η JavaFX υπερισχύει σε τεράστιο βαθμό έναντι της Swing καθώς, όχι μόνο ανταποκρίνεται στις νέες απαιτήσεις που έχουν να κάνουν με την ανάπτυξη εφαρμογών πλούσιου περιεχομένου αλλά, με τη νέα προσέγγιση του τρόπου ανάπτυξης μίας εφαρμογής, βοηθά και στο έργο του προγραμματιστή κάνοντας το ευκολότερο και δίνοντας του περισσότερες δυνατότητες τις οποίες μπορεί να ενσωματώσει στις εφαρμογές.

ΚΕΦΑΛΑΙΟ 3

Χρησιμοποιούμενες βιβλιοθήκες κώδικα για την ανάπτυξη της εφαρμογής

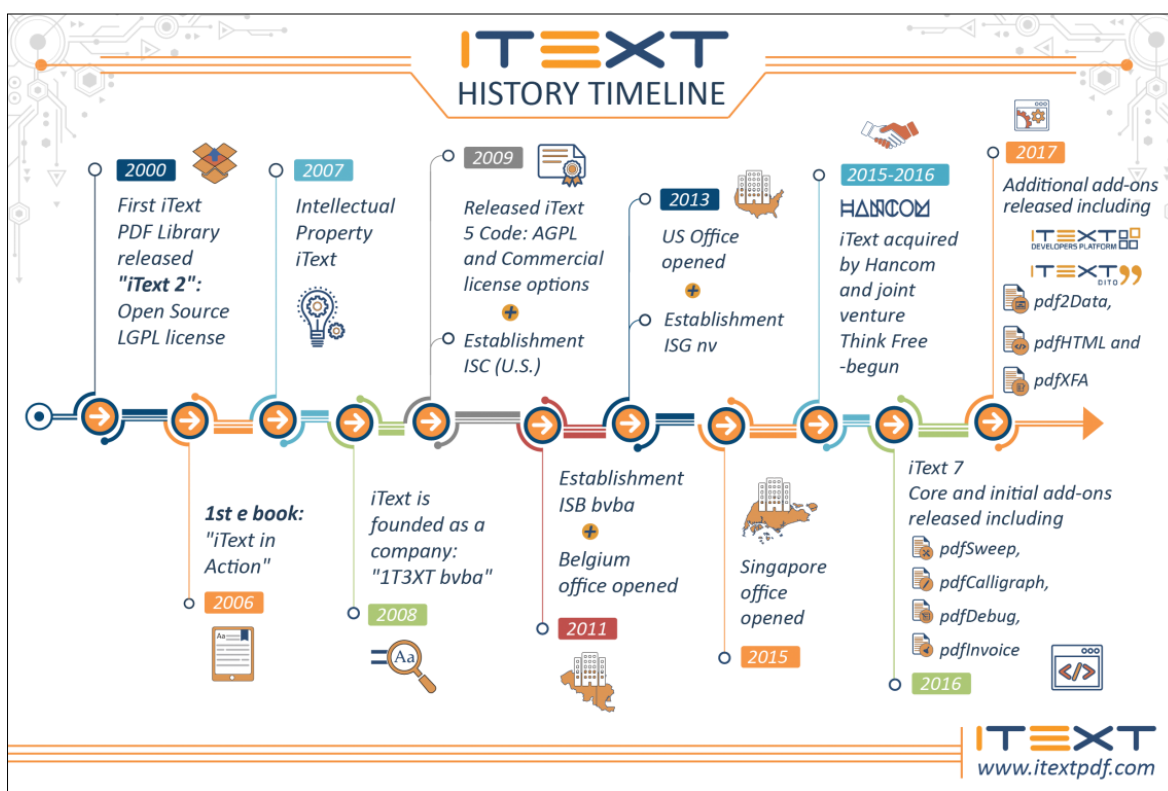
ΕΙΣΑΓΩΓΗ

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν κάποιες βιβλιοθήκες κώδικα οι οποίες βοηθούν τόσο στην υλοποίηση κάποιων βασικών δυνατοτήτων όσο και στη βελτίωση του οπτικού αποτελέσματος της εφαρμογής. Στο κεφάλαιο αυτό παρουσιάζονται οι βιβλιοθήκες αυτές κάνοντας περιγραφή των βασικών χαρακτηριστικών τους.

3.1 Η βιβλιοθήκη iText

Το iText (πρώην rugPdf) είναι μία βιβλιοθήκη ανοικτού κώδικα για τη δημιουργία και επεξεργασία αρχείων PDF. Δημιουργήθηκε το 1998 από τον Bruno Lowagie ως ενδοεπιχειρησιακό έργο στο Πανεπιστήμιο της Γάνδης.

Η αρχική μορφή του είχε απλώς τη δυνατότητα να διαβάσει, να δημιουργεί και να επεξεργάζεται PDF αρχεία, προαπαιτώντας όμως την ήδη υπάρχουσα γνώση του προγραμματιστή στη σύνταξη PDF, στα αντικείμενα, στους τελεστές κ.α. προκειμένου να υπάρχει συνεργασία με τη βιβλιοθήκη.



Εικόνα 3: Η εξέλιξη του iText

Ένα χρόνο αργότερα, ο Lowagie αποφάσισε να ξαναρχίσει από το μηδέν αφήνοντας τελείως το τότε rugPdf και δημιουργώντας το iText. Δημιουργήθηκε ως βιβλιοθήκη της Java μέσω της οποίας ο προγραμματιστής θα μπορούσε να δημιουργεί PDF έγγραφα χωρίς να γνωρίζει τη σύνταξη του PDF. Το iText κυκλοφόρησε το 2000 ως προϊόν λογισμικού ανοικτού κώδικα (FOSS)[4].

Το 2008 το iText έγινε διαθέσιμο μέσω αγοράς αδείας και το 2009 ιδρύθηκε η iText Software Corp. η οποία έγινε και ο παγκόσμιος πάροχος των προϊόντων iText. Η βιβλιοθήκη έχει μεταφερθεί και στο .NET Framework με το όνομα iTextSharp γραμμένο στη γλώσσα C#.

3.1.1 Το iText στην εφαρμογή

Η εφαρμογή της πτυχιακής “Warehouse Management” εκτός των άλλων, παρέχει τη δυνατότητα ο χρήστης να αποθηκεύει μέρος ή ολόκληρης της αποθήκης, καθώς και τις ημερήσιες πωλήσεις, σε μορφή πινάκων σε αρχεία PDF:

<i>Ημερήσιες πωλήσεις</i>						
6-01-2018						
No.	Barcode	Περιγραφή	Επωνυμία	Είδος	Ποσότητα	Ωρα
1	1001011	AMITA ΛΕΜΟΝΙ 1L	AMITA	ΑΝΑΨΥΚΤΙΚΑ-RDTS	0	15:43:49
2	1001148	TENTURA CASTRO 0.5L	CASTRO SPIRITS	ΛΙΚΕΡ	0	15:43:58
3	1001111	BELUGA GOLD LINE 0.7L	BELUGA	BOTKA	0	15:44:24
4	1001078	SCHWEPPS BITTER LEMON 0.33L	SCHWEPPS	ΑΝΑΨΥΚΤΙΚΑ-RDTS	0	15:44:35
5	1001247	HAVANA 7 YEARS OLD 0.7L	HAVANA	ΡΟΥΜΙ	0	15:44:42
6	1001169	ΒΡΥΣΣΑ ΤΣΙΠΟΥΡΟ 0.1L	ΒΡΥΣΣΑ	ΟΥΖΟ-ΤΣΙΠΟΥΡΟ-ΠΕΤΣΙΝΑ	0	15:44:48
7	1001248	HAVANA GRAND RESERVA 15 YEARS OLD	HAVANA	ΡΟΥΜΙ	0	15:45:00

Αυτό γίνεται με τη βοήθεια του iText. Πιο συγκεκριμένα, η διαδικασία δημιουργίας του αρχείου PDF χωρίζεται σε δύο σκέλη: Στη δημιουργία του εγγράφου και των ιδιοτήτων του και στη δημιουργία και εισαγωγή της πληροφορίας που αυτό θα περιέχει.

Μέρος 1^ο: Δημιουργία εγγράφου

Γενικώς, τα αρχεία PDF που δημιουργούνται από την εφαρμογή, έχει οριστεί να αποθηκεύονται σε ένα φάκελο ονόματι “Αρχεία καταγραφής αποθήκης” στην επιφάνεια εργασίας του υπολογιστή.

```
dir = new File(System.getProperty("user.home") + "\\Desktop\\Αρχεία καταγραφής αποθήκης");
if(!dir.exists()) {
    dir.mkdir();
}
```

Το όνομα του αρχείου φέρει την ημερομηνία δημιουργίας και την περιγραφή του περιεχομένου του (στην προκειμένη περίπτωση: Ημερήσιες πωλήσεις).

```
fos = new FileOutputStream(System.getProperty("user.home") +
    "\\Desktop\\Αρχεία καταγραφής αποθήκης\\Ημερήσιες πωλήσεις - " +
    (String)sdf.format(cal.getTime()) + ".pdf");
```

Για τη λήψη της τρέχουσας ημερομηνίας, χρησιμοποιούνται δύο μεταβλητές. Η πρώτη λαμβάνει την πληροφορία που φέρει την ημερομηνία και η δεύτερη ορίζει τη μορφή που θα έχει.

```
Calendar cal = Calendar.getInstance();  
SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
```

Στη συνέχεια δημιουργούνται τα fonts για τη μορφοποίηση του περιεχομένου μέσα στα πεδία του πίνακα των προϊόντων. Πιο συγκεκριμένα, δημιουργείται ένα font για τις περιγραφές των στηλών του πίνακα και ένα ακόμα για τα υπόλοιπα κελιά.

```
Font font1 = FontFactory.getFont("times.ttf", "Identity-H", true);  
font1.setSize(9);  
font1.setStyle(Font.BOLD);  
  
Font font2 = FontFactory.getFont("times.ttf", "Identity-H", true);  
font2.setSize(7);  
font2.setStyle(Font.NORMAL);
```

Με τη μέθοδο “.getFont()” ορίζουμε το font που στην προκειμένη περίπτωση δίνουμε ως όρισμα το αρχείο “times.ttf” μέσω του οποίου μπορεί να αναπαριστά ελληνικούς χαρακτήρες στη μορφή Times New Romans. Σε άλλη περίπτωση οι ελληνικοί χαρακτήρες δεν θα εμφανίζονταν καν.

Αυτό που μένει είναι να δημιουργηθεί το έγγραφο και να εισαχθεί στο παραγόμενο αρχείο. Αυτό γίνεται με τις κλάσεις Document και PdfWriter αντίστοιχα.

```
Document document = new Document(PageSize.A4, 65, 65, 100, 100);  
PdfWriter writer = PdfWriter.getInstance(document, fos);
```

Πλέον, μένει να δημιουργηθούν τα στοιχεία που πρόκειται να περιέχει το έγγραφο και να εισαχθούν σε αυτό.

Μέρος 2^ο: Δημιουργία και εισαγωγή περιεχομένου

3.1.1.1 Δημιουργία κεφαλίδας/υποσέλιδου

Εκτός του πίνακα των προϊόντων που αναπαριστά, έχει οριστεί να υπάρχει μία κεφαλίδα και ένα υποσέλιδο στην κάθε σελίδα του εγγράφου.

Προκειμένου να παρέχεται αυτή την πρόσθετη πληροφορία στο PDF έγγραφο, υπάρχει η ανάγκη να γίνει override των μεθόδων μέσω των οποίων το

iText δημιουργεί το έγγραφο. Αυτό γίνεται κληρονομώντας την κλάση “PdfPageEventHelper” η οποία περιέχει όλες τις μεθόδους που χρειάζεται το iText για τη διαδικασία αυτή.

```
@Override
public void onOpenDocument(PdfWriter writer, Document document) {
    t = writer.getDirectContent().createTemplate(30, 16);
    try {
        total = Image.getInstance(t);
        total.setRole(PdfName.ARTIFACT);
    } catch (DocumentException de) {
        throw new ExceptionConverter(de);
    }
}
```

Η μέθοδος αυτή καλείται όταν ανοίγει το έγγραφο και σε αυτή αρχικοποιούνται κάποιες βασικές μεταβλητές.

```
@Override
public void onEndPage(PdfWriter writer, Document document) {
    addHeader(writer);
    addFooter(writer);
}
```

Καλείται όταν μία σελίδα τελειώνει, ακριβώς τη στιγμή πριν η σελίδα περαστεί στο έγγραφο. Ουσιαστικά, δημιουργεί την κεφαλίδα και το υποσέλιδο με επαναληπτικό τρόπο σε κάθε σελίδα.

```
private void addHeader(PdfWriter writer) {
    PdfPTable header = new PdfPTable(2);
    try {
        // set defaults
        header.setWidths(new int[]{2, 24});
        header.setTotalWidth(527);
        header.setLockedWidth(true);
        header.getDefaultCell().setFixedHeight(40);
        header.getDefaultCell().setBorder(Rectangle.BOTTOM);
        header.getDefaultCell().setBorderColor(BaseColor.LIGHT_GRAY);

        // add image
        Image logo = Image.getInstance(getClass().getClassLoader().getResource("image.png"));
        header.addCell(logo);

        // add text
        PdfPCell text = new PdfPCell();
        text.setPaddingBottom(15);
        text.setPaddingLeft(10);
        text.setBorder(Rectangle.BOTTOM);
        text.setBorderColor(BaseColor.LIGHT_GRAY);
        text.addElement(new Phrase("Κάβα ποτών",
            FontFactory.getFont("times.ttf", "Identity-H", true, 12)));
        text.addElement(new Phrase("Διεύθυνση 10, Θεσσαλονίκη",
            FontFactory.getFont("times.ttf", "Identity-H", true, 8)));
        header.addCell(text);

        // write content
        header.writeSelectedRows(0, -1, 34, 803, writer.getDirectContent());
    } catch (Exception e) {
        System.out.println("PageEventException: " + e);
    }
}
```

Η μέθοδος αυτή είναι μία από τις πιο ουσιαστικές καθώς, σε αυτή ορίζεται το περιεχόμενο της κεφαλίδας. Το αποτέλεσμα του παραπάνω κώδικα είναι το παρακάτω:




```

private void addFooter(PdfWriter writer){
    PdfPTable footer = new PdfPTable(3);
    try {
        // set defaults
        footer.setWidths(new int[]{24, 4, 1});
        footer.setTotalWidth(527);
        footer.setLockedWidth(true);
        footer.getDefaultCell().setFixedHeight(40);
        footer.getDefaultCell().setBorder(Rectangle.TOP);
        footer.getDefaultCell().setBorderColor(BaseColor.LIGHT_GRAY);

        // add copyright
        footer.addCell(new Phrase("Πτυχιακή του Σιαμάτρα Ραφαήλ - 123929",
            FontFactory.getFont("times.ttf", "Identity-H", true, 12, Font.BOLD)));

        // add current page count
        footer.getDefaultCell().setHorizontalAlignment(Element.ALIGN_RIGHT);
        footer.addCell(new Phrase(String.format("Σελίδα %d από", writer.getPageNumber()),
            FontFactory.getFont("times.ttf", "Identity-H", true, 8)));

        // add placeholder for total page count
        PdfPCell totalPageCount = new PdfPCell(total);
        totalPageCount.setBorder(Rectangle.TOP);
        totalPageCount.setBorderColor(BaseColor.LIGHT_GRAY);
        footer.addCell(totalPageCount);

        // write page
        PdfContentByte canvas = writer.getDirectContent();
        canvas.beginMarkedContentSequence(PdfName.ARTIFACT);
        footer.writeSelectedRows(0, -1, 34, 50, canvas);
        canvas.endMarkedContentSequence();
    } catch(DocumentException de) {
        throw new ExceptionConverter(de);
    }
}

```

Η δεύτερη βασική μέθοδος είναι η παραπάνω μέσω της οποίας δημιουργείται το υποσέλιδο για κάθε σελίδα του εγγράφου. Παρακάτω παρουσιάζεται το αποτέλεσμα:

```
@Override
public void onCloseDocument(PdfWriter writer, Document document) {
    int totalLength = String.valueOf(writer.getPageNumber()).length();
    int totalWidth = totalLength * 5;
    ColumnText.showTextAligned(t, Element.ALIGN_RIGHT,
        new Phrase(String.valueOf(writer.getPageNumber()),
            FontFactory.getFont("times.ttf", "Identity-H", true, 8)),
            totalWidth, 6, 0);
}
```

Η μέθοδος αυτή καλείται ακριβώς πριν κλείσει το έγγραφο. Σε αυτό το σημείο είναι που υπολογίζεται το σύνολο των σελίδων που δίνεται στο template που ορίζεται στην πρώτη μέθοδο.

3.1.1.2 Δημιουργία πίνακα

Από τη στιγμή που δημιουργηθούν και η κεφαλίδα με το υποσέλιδο, μένει να αναπαρασταθεί η βασική πληροφορία του εγγράφου, η οποία είναι ο πίνακας με τα ζητούμενα προϊόντα.

```
PdfPTable pt = new PdfPTable(widths);
pt.setTotalWidth(500);
pt.setLockedWidth(true);
pt.setHorizontalAlignment(Element.ALIGN_CENTER);
```

Αρχικά, δημιουργείται ο πίνακας τύπου PdfPTable του οποίου ο δομητής δέχεται ως όρισμα τις στήλες που θα περιέχει.

```
float[] widths = {0.04f, 0.08f, 0.31f, 0.23f, 0.15f, 0.10f, 0.09f};
```

Η μεταβλητή widths περιέχει μία τιμή, για την κάθε στήλη, που αναπαριστά το μήκος της κάθε μίας.

```
for (int i=0; i<metadata.length; i++) {
    cell = new PdfPCell(new Phrase(metadata[i], font1));
    cell.setHorizontalAlignment(Element.ALIGN_CENTER);
    cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    pt.addCell(cell);
}
```

Στην συνέχεια ο πίνακας συμπληρώνεται με τις αντίστοιχες τιμές. Αρχικά, συμπληρώνονται τα μεταδεδομένα του πίνακα, δηλαδή το όνομα της κάθε στήλης. Η μέθοδος `addCell` του `iText` προσθέτει ένα κελί με προτεραιότητα τη συμπλήρωση μίας σειράς.

Η μεταβλητή `metadata` περιλαμβάνει σε μορφή `Strings` τα ονόματα όλων των στηλών.

```
int size = salesTable.getItems().size();
int columns = salesTable.getColumns().size();

String data;

for(int i=0; i<size; i++) {
    for(int j=0; j<columns; j++) {
        if (salesTable.getColumns().get(j).getCellData(i) != null) {
            data = salesTable.getColumns().get(j).getCellData(i).toString();
        }
        else {
            data = "-";
        }

        phrase = new Phrase(data, font2);
        cell = new PdfPCell(phrase);
        cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);

        pt.addCell(cell);
    }
}
```

Έπειτα, συμπληρώνεται ο υπόλοιπος πίνακας με τα ζητούμενα προϊόντα. Στην προκειμένη περίπτωση έχουμε τις ημερήσιες πωλήσεις οι οποίες είναι αποθηκευμένες στον πίνακα `salesTable`.

Η κάθε μία τιμή περνά στη μεταβλητή `data`. Μέσω της κλάσης `Phrase` δημιουργείται μία φράση η οποία αποτελείται από τα `data` και από το `font` που θα έχει. Το αντικείμενο `phrase` περνά στη μεταβλητή `cell` η οποία είναι τύπου `PdfPCell` και αυτή με τη σειρά της θα προστεθεί τελικά στον πίνακα του εγγράφου. Η διαδικασία αυτή επαναλαμβάνεται για όλα τα στοιχεία που βρίσκονται στον πίνακα των ημερήσιων πωλήσεων.

3.1.1.3 Δημιουργία κεφαλίδων εγγράφου

Ένα άλλο στοιχείο του εγγράφου είναι οι κεφαλίδες που περιέχει. Στο iText ένα τρόπο για να προστεθούν είναι μέσω της κλάσης “Paragraph”.

```
Paragraph p1 = new Paragraph("Ημερήσιες πωλήσεις",  
    FontFactory.getFont("times.ttf", "Identity-H", true, 24, Font.BOLDITALIC));  
p1.setAlignment(Element.ALIGN_CENTER);  
  
Paragraph p2 = new Paragraph((String) sdf.format(cal.getTime()),  
    new Font(Font.FontFamily.TIMES_ROMAN, 16));  
p2.setAlignment(Element.ALIGN_CENTER);
```

Το αποτέλεσμα του παραπάνω κώδικα παρουσιάζεται παρακάτω:



Ημερήσιες πωλήσεις
6-01-2018

3.1.1.4 Εισαγωγή των στοιχείων στο έγγραφο

Τέλος, τα στοιχεία που έχουν δημιουργηθεί όπως ο πίνακας και οι κεφαλίδες χρειάζεται να προστεθούν στο έγγραφο.

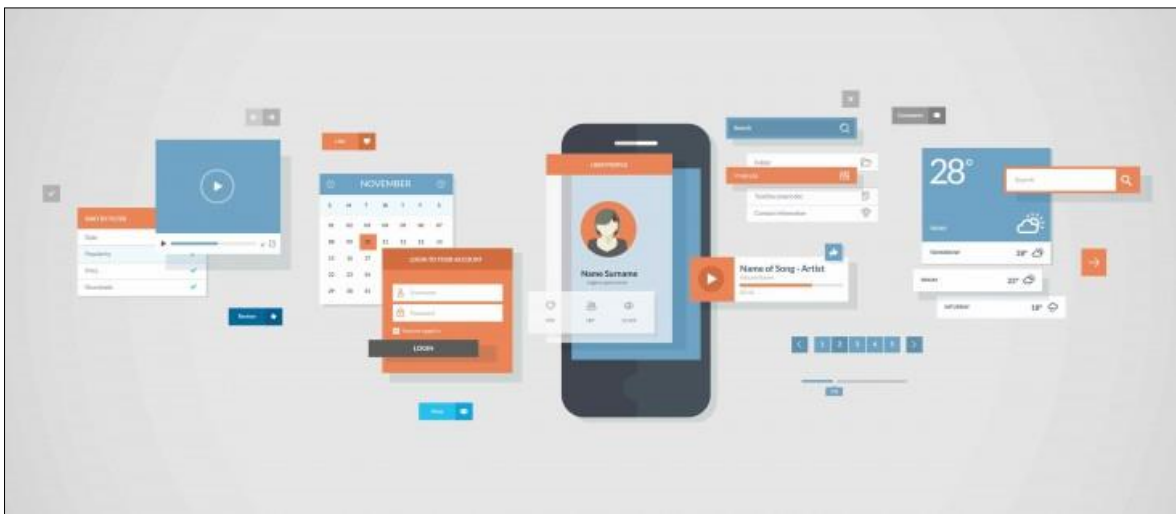
```
document.open();  
document.addAuthor("Warehouse Management");  
document.addTitle("Ημερήσιες πωλήσεις");  
document.add(Chunk.NEWLINE);  
document.add(p1);  
document.add(p2);  
document.add(Chunk.NEWLINE);  
document.add(pt);  
document.close();
```

Η κλάση “Document” περιέχει διάφορες μεθόδους μέσω των οποίων ο προγραμματιστής μπορεί να προσθέσει στο PDF αρχείο, τίτλο, συγγραφέα, ημερομηνία και φυσικά όλα τα στοιχεία που δομούν το έγγραφο.

3.2 Η βιβλιοθήκη γραφικών JFoenix

Η JavaFX όπως έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο, έχει αναπτύξει σε μεγάλο βαθμό το θέμα συστατικών στοιχείων της διεπαφής χρήστη. Υπάρχει μία αρκετά μεγαλύτερη πληθώρα στοιχείων σε σύγκριση με την Swing, καθώς επίσης, δίνει και τη δυνατότητα στον προγραμματιστή, να ορίζει το layout αυτών, όπως αυτός θέλει σε οποιαδήποτε χρονική στιγμή κατά την οποία η εφαρμογή αλληλεπιδρά με το χρήστη.

Ωστόσο, η εμφάνιση των στοιχείων αυτών, κατά πολλούς, δεν είναι τόσο ελκυστική. Λύση σε αυτό έδωσε η Google μέσω του Material design.



Εικόνα 4: Το Material Design της Google

Ένα σύνολο σχεδιαστικών κανόνων, βασισμένο σε διάταξη πίνακα (grid layout), με κινούμενα σχέδια και μεταβάσεις που ανταποκρίνονται στις κινήσεις των χρηστών, χρήση padding και εφέ βάθους με κατάλληλο φωτισμό και σκίαση των αντικειμένων. Πλέον, το Material design έχει εφαρμοστεί σε όλες τις εφαρμογές της Google όπως οι Gmail, YouTube, Google Drive κ.α.[5].

Στο σημείο αυτό έρχεται η ομάδα του JFoenix η οποία ανέλαβε να μεταφέρει το Material Design και στην Java. Το JFoenix είναι το αποτέλεσμα αυτής της ομάδας, καθώς είναι η βιβλιοθήκη η οποία τα περιέχει όλα αυτά.

Το μόνο που έχει να κάνει ο προγραμματιστής ο οποίος θέλει να προσθέσει τα components της Google, είναι να προσθέσει τη βιβλιοθήκη του JFoenix στην εφαρμογή του και στο Scene Builder στον οποίο θα δημιουργήσει τη διεπαφή χρήστη της εφαρμογής[6].

Τα components του JFoenix, από τη στιγμή που βασίζονται πάνω σε αυτά της JavaFX, υποστηρίζουν τη δυνατότητα τροποποίησης τους μέσω του CSS.

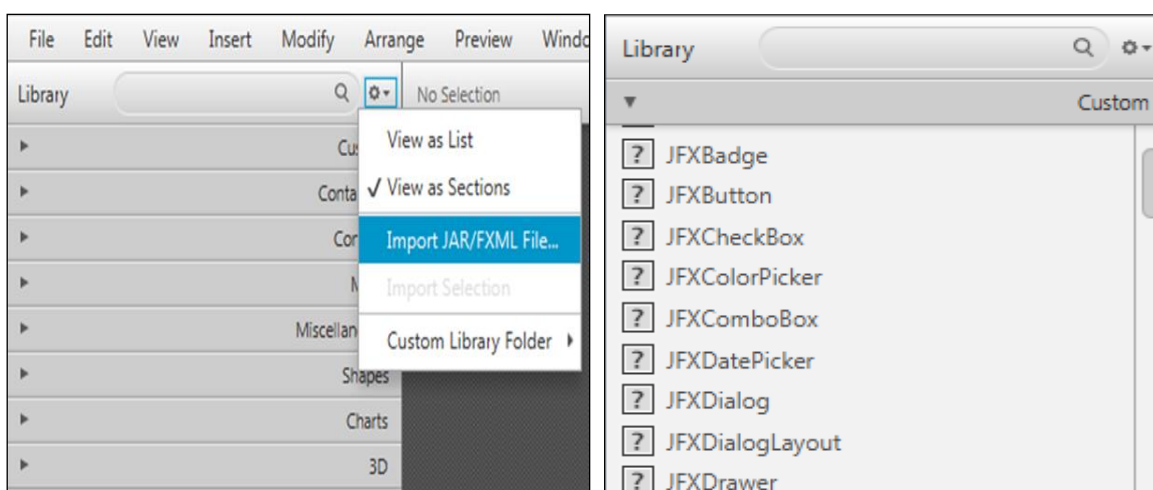
3.2.1 Λήψη και εγκατάσταση του JFoenix

Το JFoenix είναι διαθέσιμο προς λήψη από την επίσημη ιστοσελίδα <http://www.jfoenix.com/> η οποία με τη σειρά της σε παραπέμπει για τη λήψη του αρχείου, στη γνωστή ιστοσελίδα διαμοιρασμού κώδικα, GitHub.



Εικόνα 5: Η επίσημη ιστοσελίδα του JFoenix

Προκειμένου ο προγραμματιστής να έχει στη διάθεση του τα components, θα πρέπει να προσθέσει τη βιβλιοθήκη στο Scene Builder. Αυτό γίνεται με τρόπο αντίστοιχο αυτού που προσθέτει κάποιος μία νέα βιβλιοθήκη στα γνωστότερα IDE.

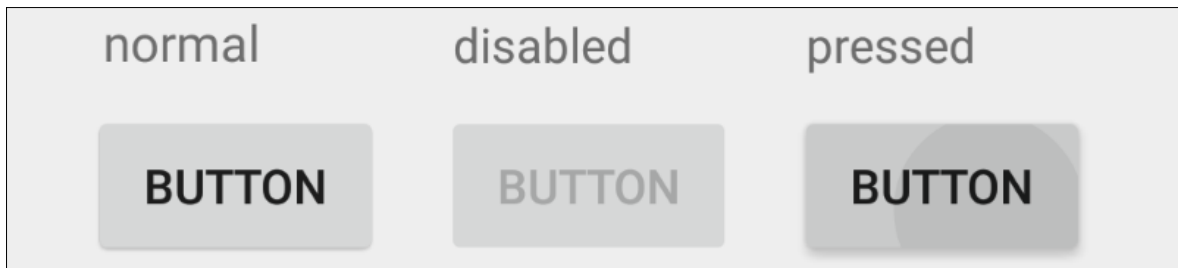


Εάν η εγκατάσταση της βιβλιοθήκης ολοκληρωθεί επιτυχώς, τα νέα components θα είναι πλέον διαθέσιμα στην παλέτα μαζί με τα υπόλοιπα στοιχεία.

3.2.2 Το JFoenix στην εφαρμογή

Στην εφαρμογή της πτυχιακής έχουν χρησιμοποιηθεί δύο είδη component του JFoenix. Πιο συγκεκριμένα, τα JFXButton και JFXTextField.

Η αρχική μορφή των JFXButtons την οποία έχουν από το Design material της Google είναι η εξής:



Ωστόσο, και μέσω του CSS η τελική μορφή αυτών στην εφαρμογή, είναι η παρακάτω:



Η τροποποίηση των στοιχείων του JFoenix μέσω του CSS γίνεται με τρόπο αντίστοιχο αυτού που τροποποιούνται και τα στοιχεία της JavaFX με τη μόνη διαφορά ότι τα πρώτα φέρουν το πρόθεμα "jfx-".

```
.jfx-combo-box {  
    -fx-prompt-text-fill:#2c7db4;  
}
```

Τα JFXTextFields είναι αυτά που δίνουν την εντύπωση ενός πιο 'ζωντανού' στοιχείου με το οποίο ο χρήστης νιώθει εντονότερα την αλληλεπίδραση του με αυτό. Οι σκιές, η τρισδιάστατη μορφή, η εναλλαγή των χρωμάτων, σε συνεργασία τη δυνατότητα να ελέγχεις τις ιδιότητες του component μέσω της JavaFX, είναι ο τρόπος με τον οποίο επιτυγχάνεται αυτό.



3.3 Η βάση δεδομένων και η επικοινωνία με αυτή

Η εφαρμογή βασίζεται καθαρά πάνω σε μία βάση δεδομένων, η οποία περιέχει όλα τα δεδομένα για τη διαχείριση της αποθήκης. Η συγκεκριμένη βάση της πτυχιακής ανήκει στην PostgreSQL.

Στην παρούσα φάση η εφαρμογή εστιάζει αποκλειστικά στη διαχείριση των προϊόντων της αποθήκης. Για το λόγο αυτό, το μόνο αντικείμενο το οποίο διατηρείται στη βάση είναι τα προϊόντα. Έτσι, η βάση περιέχει έναν πίνακα με όλα τα προϊόντα. Κάθε προϊόν έχει τις παρακάτω ιδιότητες και το σχήμα περιλαμβάνει τους παρακάτω κανόνες:

- **Κωδικός:** String 7 χαρακτήρων, κύριο κλειδί του πίνακα (not null).
- **Περιγραφή:** String 1-50 χαρακτήρων (not null).
- **Επωνυμία:** String 1-30 χαρακτήρων.
- **Είδος:** String 1-30 χαρακτήρων (not null).
- **Απόθεμα:** Integer
- **Φωτογραφία:** bytea

Από τη μεριά της εφαρμογής, η εισαγωγή νέων προϊόντων περιορίζεται στους εξής κανόνες:

- Όλοι οι χαρακτήρες είναι άτονοι και κεφαλαίοι.
- Ο κωδικός περιέχει μόνο αριθμητικούς χαρακτήρες.
- Επιτρεπόμενοι ειδικοί χαρακτήρες είναι οι: *,.&-'%.

Για να πραγματοποιηθεί η σύνδεση στη βάση και να υπάρχει μία αμφίδρομη επικοινωνία μεταξύ βάσης δεδομένων και εφαρμογής, είναι απαραίτητη η ύπαρξη ενός Java API το οποίο παρέχει τις κατάλληλες μεθόδους και τους κανόνες για την επικοινωνία αυτή. Το API αυτό ονομάζεται JDBC.

3.4 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν οι βιβλιοθήκες οι οποίες χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Επίσης, έγινε περιγραφή του τρόπου με τον οποίο έγινε η εφαρμογή αυτών. Από τη μία, η iText παρέχει ένα σημαντικό εργαλείο στο προγραμματιστή, την ικανότητα της εφαρμογής να δημιουργεί PDF αρχεία τα οποία έχουν τη μορφή και το περιεχόμενο που έχει ορίσει ο προγραμματιστής. Από την άλλη, το JFoenix μπορεί να δώσει στη διεπαφή μία πιο ζωντανή απεικόνιση των στοιχείων της, βελτιστοποιώντας παράλληλα την αίσθηση που έχει ο χρήστης, όσον αφορά την αλληλεπίδραση του με την εφαρμογή.

ΚΕΦΑΛΑΙΟ 4

Η εφαρμογή “Warehouse Management”

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό γίνεται μία ολοκληρωμένη περιγραφή της εφαρμογής η οποία αποτελεί και το αντικείμενο της παρούσας πτυχιακής εργασίας. Στα επόμενα υποκεφάλαια γίνεται παρουσίαση των συνολικά τεσσάρων μερών της εφαρμογής. Σε κάθε μέρος, θα γίνεται αρχικά περιγραφή της διεπαφής χρήστη, έπειτα περιγραφή του κώδικα και τέλος περιγραφή του CSS της διεπαφής.

Σκοπός της εφαρμογής είναι, ο χρήστης της να μπορεί να διαχειρίζεται τα αποθέματα των προϊόντων μίας αποθήκης ενός καταστήματος, καθώς και τις ημερήσιες πωλήσεις του καταστήματος. Πιο συγκεκριμένα, ο χρήστης μπορεί να προσθέτει νέα προϊόντα, να επεξεργάζεται ή να αφαιρεί ήδη υπάρχοντα καθώς και να ενημερώνει την αποθήκη με το νέο απόθεμα από το προϊόν που πωλείται κάθε φορά. Για την αφαίρεση του πωληθέντος προϊόντος, μπορεί να χρησιμοποιηθεί ένα barcode Scanner, το οποίο διαβάζει απευθείας του barcode του προϊόντος.

Το βασικά χαρακτηριστικά της εφαρμογής ωστόσο είναι, ο έλεγχος που γίνεται στην εισαγωγή των δεδομένων από το χρήστη, η απλούστευση στο οπτικό αποτέλεσμα που βλέπει τελικά ο χρήστης, καθώς και η ενημέρωση της εφαρμογής προς το χρήστη μέσω ειδοποιητικών μηνυμάτων σχεδόν σε κάθε του κίνηση, ούτως ώστε να είναι σίγουρος για την κάθε του ενέργεια.

Στο τέλος του κεφαλαίου γίνονται προτάσεις για εναλλακτικούς τρόπους υλοποίησης κάποιων δυνατοτήτων της εφαρμογής, καθώς και νέες δυνατότητες οι οποίες θα μπορούσαν να προστεθούν μελλοντικά στην εφαρμογή.

4.1 Περιγραφή της Διεπαφής Χρήστη

4.1.1 Κύρια διεπαφή – Προεπισκόπηση της αποθήκης

Η αρχική σελίδα της εφαρμογής περιέχει στο βασικό της μέρος, ένα πίνακα με όλα τα προϊόντα της αποθήκης με όλα τους τα χαρακτηριστικά (Κωδικός, Περιγραφή, Επωνυμία, Είδος και Απόθεμα). Υπάρχει η δυνατότητα να γίνει ταξινόμηση των προϊόντων ανάλογα με το πεδίο που θέλει ο χρήστης.

Στο αριστερό μέρος υπάρχει ένα πεδίο με το οποίο ο χρήστης μπορεί να κάνει αναζήτηση ένα ή περισσότερα προϊόντα που εμπεριέχουν τη φράση αναζήτησης στο Κωδικό, στην Περιγραφή ή στην Επωνυμία του προϊόντος. Τα αποτελέσματα εμφανίζονται στον ήδη υπάρχοντα πίνακα και ανανεώνονται κάθε φορά που ο χρήστης πληκτρολογεί ένα χαρακτήρα στο πεδίο αναζήτησης.

The screenshot shows the 'Warehouse Management' application. On the left, there is a sidebar with a search bar 'Αναζήτηση προϊόντος...', a 'Στοιχεία προϊόντος' section with a 'NO IMAGE AVAILABLE' placeholder, and a list of product details for 'FANTA ΠΟΡΤΟΚΑΛΑΔΑ ΜΠΛΕ 330ΜL'. The main area is a table with the following columns: Κωδικός, Περιγραφή, Επωνυμία, Είδος, and Απόθεμα. The table lists various products including ALOE VERA, AMITA, and COCA COLA.

Κωδικός	Περιγραφή	Επωνυμία	Είδος	Απόθεμα
1001001	ALOΕ VERA ΚΟΥΤΙ ΚΛΑΣΣΙΚΟ 240ML	ALOΕ VERA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001002	ALOΕ VERA ΚΟΥΤΙ ΡΟΔΙ 240ML	ALOΕ VERA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001003	ALOΕ VERA ΚΟΥΤΙ GUAVA 240ML	ALOΕ VERA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001004	ΑΜΙΤΑ ΜΟΤΙΟΝ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001005	ΑΜΙΤΑ ΜΟΤΙΟΝ 330ML	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001006	ΑΜΙΤΑ ΑΝΑΝΑΣ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001007	ΑΜΙΤΑ ΒΥΣΙΝΟ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001008	ΑΜΙΤΑ ΒΥΣΙΝΟ 250ML	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001009	ΑΜΙΤΑ ΜΗΛΟ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001010	ΑΜΙΤΑ ΜΠΑΝΑΝΑ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001011	ΑΜΙΤΑ ΛΕΜΟΝΙ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001012	ΑΜΙΤΑ ΠΙΜΒ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001013	ΑΜΙΤΑ ΝΕΚΤΑΡ ΜΠΛΕ 0.25L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001014	ΑΜΙΤΑ ΠΙΜΚ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001015	ΑΜΙΤΑ ΡΟΔΑΚΙΝΟ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001016	ΑΜΙΤΑ ΡΟΔΑΚΙΝΟ 250ML	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001017	ΑΜΙΤΑ ΡΟΔΙ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001018	ΑΜΙΤΑ ΝΕΣΤΑΡ ΦΥΣΙΚΟΣ ΧΥΜΟΣ ΠΟΡΤΟΚΑΛΙ 1L	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001019	ΑΜΙΤΑ ΦΥΣΙΚΟΣ ΧΥΜΟΣ ΠΟΡΤΟΚΑΛΙ 330ML	ΑΜΙΤΑ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001020	ΗΒΗ ΠΟΡΤΟΚΑΛΙ 250ML	ΗΒΗ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001021	ΗΒΗ GO ΜΗΛΟ ΠΟΡΤΟΚΑΛΙ ΜΑΝΤΑΡΙΝΙ 250ML	ΗΒΗ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001022	ΗΒΗ GO ΜΗΛΟ ΠΟΡΤΟΚΑΛΙ ΡΟΔΑΚΙΝΟ ΒΕΡΙΚΟΚΟ 250ML	ΗΒΗ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001023	ΗΒΗ GO ΦΡΑΟΥΛΑ ΜΗΛΟ ΣΤΑΦΥΛΙ 250ML	ΗΒΗ	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001024	BACARDI BREEZER ΛΕΜΟΝΙ-ΛΑΙΜ-ΚΑΠΡΟΥΖΙ-ΑΝΑΝΑΣ 0.275L	BACARDI	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001025	COCA COLA ΠΛΑΣΤΙΚΟ 0.5L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001026	COCA COLA 1.5L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001027	COCA COLA 1L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001028	COCA COLA ΚΟΥΤΙ 0.33L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001029	COCA COLA LIGHT 0.5L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001030	COCA COLA LIGHT 1.5L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001031	COCA COLA LIGHT ΚΟΥΤΙ 0.33L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0
1001032	COCA COLA ZERO 0.5L	COCA COLA	ΑΝΑΨΥΚΤΙΚΑ-RDT'S	0

Εικόνα 6: Η κεντρική σελίδα της εφαρμογής

Κάτω από το πεδίο αναζήτησης, υπάρχουν τα πεδία τα οποία συμπληρώνονται με τα χαρακτηριστικά του προϊόντος που επιλέγεται κάθε φορά από τον πίνακα. Ο χρήστης έχει τη δυνατότητα να επεξεργαστεί από το αντίστοιχο πεδίο, το απόθεμα του προϊόντος. Αφού γίνει αυτό, με το κουμπί «Αποθήκευση» μπορεί να ενημερώσει το νέο απόθεμα του προϊόντος. Όπως επίσης, έχει τη δυνατότητα μέσω του κουμπιού «Διαγραφή προϊόντος» να διαγράψει το προϊόν από τη βάση.

Η κύρια σελίδα παρέχει επίσης ένα μενού μέσω του οποίου μπορεί ο χρήστης να χρησιμοποιήσει τις υπόλοιπες δυνατότητες της εφαρμογής:

- **Αποθήκη → Ανανέωση:** Ανανεώνει τον πίνακα με τα προϊόντα για τυχόν αλλαγές στη βάση.
- **Αποθήκη → Καταχώρηση νέου προϊόντος...:** Ανοίγει παράθυρο για την καταχώρηση νέου προϊόντος στη βάση.
- **Αποθήκη → Εκτύπωση αποθήκης:** Υπάρχει η δυνατότητα ο χρήστης να δημιουργήσει ένα αρχείο PDF το οποίο μπορεί αργότερα να εκτυπώσει. Το αρχείο αυτό μπορεί να περιέχει σε μορφή πίνακα:
 1. Το σύνολο των προϊόντων της αποθήκης.
 2. Τα προϊόντα της αποθήκης με μη μηδενικό απόθεμα.
 3. Τα προϊόντα της αποθήκης με μηδενικό απόθεμα.
- **Πωλήσεις → Ημερήσιες πωλήσεις:** Ανοίγει παράθυρο για τη διαχείριση των ημερήσιων πωλήσεων.
- **Εφαρμογή → Σχετικά με την εφαρμογή...:** Ανοίγει παράθυρο με πληροφορίες σχετικές με την εφαρμογή.
- **Εφαρμογή → Έξοδος:** Μετά από επιβεβαίωση του χρήστη, τερματίζεται η εφαρμογή.

4.1.2 Κύρια διεπαφή - Δημιουργία προϊόντος

Μέσω της διεπαφής αυτής, ο χρήστης έχει τη δυνατότητα να δημιουργήσει ένα νέο προϊόν. Μεγάλη βαρύτητα όπως είναι προφανές, πέφτει στην εγκυρότητα των δεδομένων τα οποία εισάγει ο χρήστης. Γι' αυτό το λόγο, η εφαρμογή επιτρέπει ή απαγορεύει κάποια χαρακτηριστικά:

- Δημιουργία νέου είδους προϊόντος σε περίπτωση που αυτό δεν υπάρχει ήδη στη βάση.
- Εισαγωγή συγκεκριμένων ειδικών χαρακτήρων στην Περιγραφή, την Επωνυμία και το είδος (*,.&-'%).
- Εισαγωγή μόνο αριθμητικών χαρακτήρων στο πεδίο του αποθέματος (0-999999).
- Εισαγωγή φωτογραφίας προϊόντος (συγκεκριμένοι τύποι αρχείων).

Warehouse Management

Καταχώρηση νέου προϊόντος

Είδος

Νέο είδος

Περιγραφή -Μέχρι 50 χαρακτήρες.

Επωνυμία -Μέχρι 30 χαρακτήρες.

Κωδικός
1001281

Απόθεμα

Αποθήκευση Ακύρωση

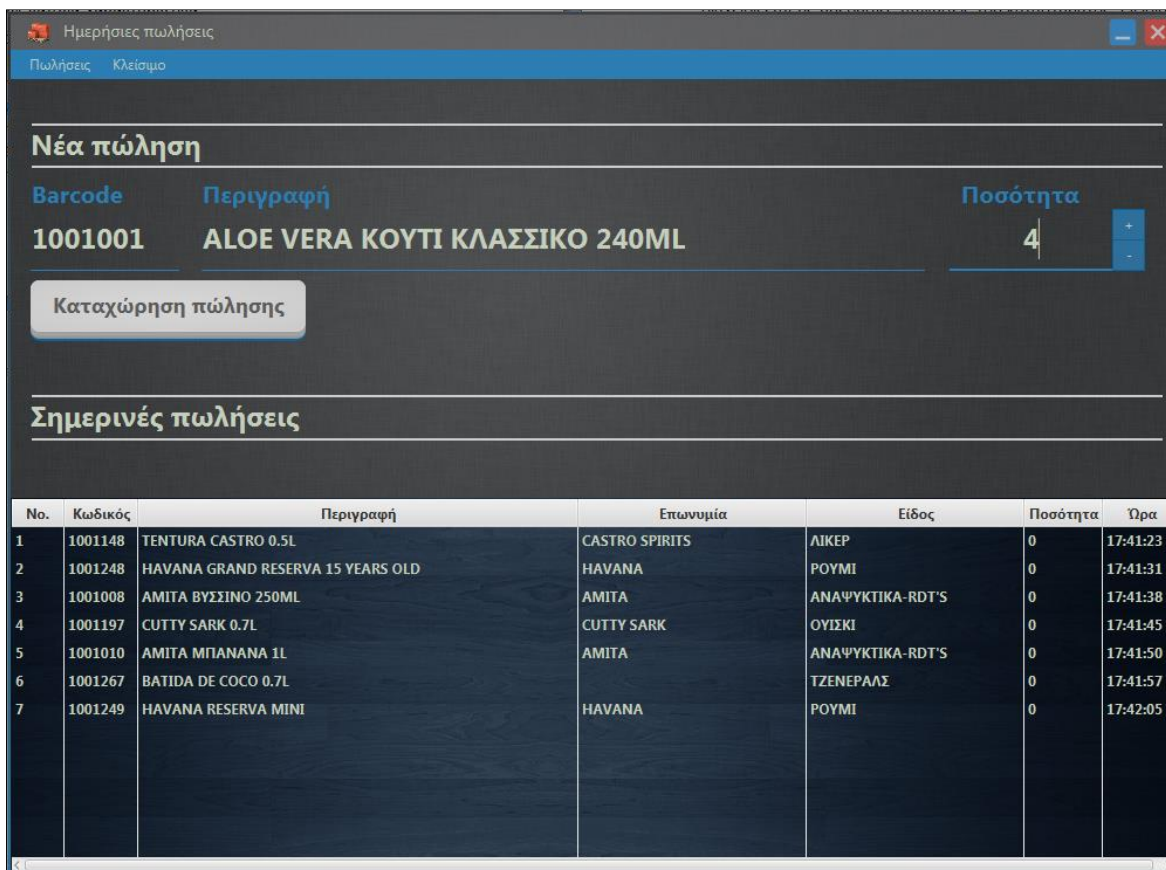
* Υποχρεωτικό πεδίο

Εικόνα 7: Φόρμα δημιουργίας νέου προϊόντος

Ο κωδικός του νέου προϊόντος, δημιουργείται αυτόματα. Αυτό γίνεται για λόγους αξιοποίησης όλων των κωδικών που ανήκουν στο εύρος που έχει οριστεί. Πιο συγκεκριμένα, επειδή ο χρήστης δε μπορεί να γνωρίζει ποιος κωδικός είναι ο αμέσως επόμενος που είναι διαθέσιμος, καθώς κάποιοι ενδιαμέσοι κωδικοί ενδέχεται να έχουν ελευθερωθεί λόγω της διαγραφής του προϊόντος τους, η εφαρμογή βρίσκει τον αμέσως επόμενο διαθέσιμο κωδικό και τον συμπληρώνει στο αντίστοιχο πεδίο.

4.1.3 Κύρια διεπαφή - Ημερήσιες πωλήσεις

Η τρίτη διεπαφή έχει δημιουργηθεί προκειμένου ο χρήστης να μπορεί να διαχειρίζεται τις ημερήσιες πωλήσεις του καταστήματος. Οι ιδιότητες των στοιχείων της διεπαφής έχουν παραμετροποιηθεί με τέτοιο τρόπο ούτως ώστε να διευκολύνεται ο χρήστης στην εισαγωγή του προϊόντος και να ελαχιστοποιείται η διάρκεια της διαδικασίας εισαγωγής.



Εικόνα 8: Διαχείριση των ημερήσιων πωλήσεων

Ο χρήστης, αυτό που έχει να κάνει είναι να εισάγει τον κωδικό του προϊόντος, η οποία διαδικασία επιταχύνεται σε μεγάλο βαθμό με τη χρήση barcode scanner, και την ποσότητα του προς πώληση προϊόντος. Με τη συμπλήρωση του κωδικού, σε περίπτωση που αυτός αντιστοιχεί σε κάποιο προϊόν της βάσης, η εφαρμογή συμπληρώνει αυτόματα την περιγραφή του προϊόντος στο αντίστοιχο πεδίο. Ο χρήστης τότε έχει τη δυνατότητα να επιβεβαιώσει την ορθότητα των δεδομένων και να ολοκληρώσει την πώληση. Το πωληθέν προϊόν τότε, προστίθεται στον πίνακα των ημερήσιων πωλήσεων με μία επιπρόσθετη πληροφορία, την ώρα που πραγματοποιήθηκε η πώληση.

Όπως και στην κύρια σελίδα, όπου υπάρχει η δυνατότητα εκτύπωσης μέρους ή ολόκληρης της αποθήκης, έτσι και εδώ, ο χρήστης έχει τη δυνατότητα να αποθηκεύσει ή και να εκτυπώσει σε αρχείο PDF τις πωλήσεις της συγκεκριμένης ημέρας.

4.1.4 Δευτερεύουσα διεπαφή - Σχετικά με την εφαρμογή

Η τελευταία διεπαφή παρέχει μία σύντομη περιγραφή της εφαρμογής και κάποιες πληροφορίες για το δημιουργό της εφαρμογής. Σε αυτή τη διεπαφή έχει

γίνει χρήση κάποιων στοιχείων της JavaFX και πιο συγκεκριμένα των Hyperlink και Tooltip, τα οποία αναλύονται παρακάτω στην περιγραφή του κώδικα της εφαρμογής.



Εικόνα 9: Παράθυρο περιγραφής της εφαρμογής

4.2 Περιγραφή του κώδικα της εφαρμογής

Στο σημείο αυτό γίνεται περιγραφή κάποιων σημείων του κώδικα, είτε για την κατανόηση των βασικών μεθόδων για τη λειτουργία μίας JavaFX εφαρμογής, είτε για την παρουσίαση κάποιων μεθόδων που επικεντρώνονται στη διαχείριση των ιδιοτήτων των συστατικών στοιχείων της JavaFX, αλλά και στους ακροατές γεγονότων, που ως σκοπό έχουν την καλύτερη και αποτελεσματικότερη αλληλεπίδραση με το χρήστη.

Τυχαίνει σε αρκετά σημεία του κώδικα, να επαναλαμβάνεται η ίδια μεθοδολογία για την επίλυση διαφορετικών ζητημάτων. Ωστόσο, για λόγους σκοπιμότητας, κάθε μεθοδολογία παρουσιάζεται μία φορά.

4.2.1 Βασικά βήματα για τη λειτουργία μίας JavaFX εφαρμογής

Αρχικά, χρειάζεται να δημιουργηθεί το Stage, μέσα στο οποίο βρίσκονται όλα τα στοιχεία της JavaFX. Αυτό γίνεται μέσα στη μέθοδο ονόματι start.

```
@Override
public void start(Stage primaryStage) throws Exception {

    Parent root = FXMLLoader.load(getClass().getResource("Main.fxml"));

    Scene scene = new Scene(root);
    scene.getStylesheets().add(getClass().getResource("mainCSS.css").toExternalForm());
    mainStage = new Stage();
    mainStage.getIcons().add(new Image("image.png"));
    mainStage.setTitle("Warehouse Management");
    mainStage.setScene(scene);
    mainStage.initStyle(StageStyle.UNDECORATED);
    mainStage.setOnCloseRequest((WindowEvent event) -> {
        event.consume();
        closing();
    });
    mainStage.show();

    //checkQuantity();
}
```

Οι γραμμές με το κόκκινο στίγμα είναι οι απαραίτητες για τη λειτουργία της εφαρμογής. Παρακάτω γίνεται περιγραφή των εντολών αυτών με σειρά εμφάνισης:

1. Ως αντικείμενο root ορίζεται η διεπαφή χρήστη η οποία δημιουργήθηκε και αποθηκεύτηκε στο αρχείο "Main.fxml".
2. Στη σκηνή περνά ως όρισμα η διεπαφή που βρίσκεται πλέον στο αντικείμενο root.
3. Αρχικοποιείται το αντικείμενο τύπου Stage.
4. Η σκηνή περνά ως όρισμα στο stage.
5. Γίνεται εμφάνιση του παραθύρου.

Προκειμένου η εφαρμογή να ξεκινήσει χρειάζεται να τρέξει η κύρια μέθοδος:

```
public static void main(String[] args) {
    launch(args);
}
```


Από κει και πέρα, μέσω άλλων μεθόδων οι ιδιότητες του stage και οι πληροφορίες που αυτό θα φέρει, μπορούν να παραμετροποιηθούν με τις ανάλογες μεθόδους. Παρακάτω γίνεται περιγραφή των δευτερευουσών μεθόδων:

1. Με τη μέθοδο `getStylesheets` ορίζουμε ένα αρχείο τύπου `css` σύμφωνα με το οποίο θα μορφοποιηθούν τα στοιχεία της σκηνής.
2. Με τη μέθοδο `getIcons` μπορεί να προστεθεί το εικονίδιο που θα φέρει η εφαρμογή.
3. Με τη μέθοδο `setTitle` ορίζεται ο τίτλος της εφαρμογής.
4. Με την εντολή `mainStage.initStyle(StageStyle.UNDECORATED)` ορίζεται πως το συγκεκριμένο stage δε θα φέρει το πλαίσιο παραθύρου του λειτουργικού συστήματος στο οποίο η εφαρμογή τρέχει. Ο λόγος για τον οποίο γίνεται είναι καθαρά πειραματικός, καθώς έχει αναπτυχθεί σε αρχικό στάδιο ένα προσαρμοσμένο στυλ πλαισίου.
5. Στη μέθοδο `setOnCloseRequest`, η οποία ενεργοποιείται όταν δίνεται εντολή να κλείσει το παράθυρο, ορίζεται ένα `expression Lambda` το οποίο προωθεί τη ροή του προγράμματος σε μία άλλη μέθοδο αποτρέποντας το κλείσιμο.

Ένα δεύτερο πολύ βασικό στοιχείο είναι η δημιουργία της κλάσης, η οποία θα περιέχει όλες εκείνες τις μεθόδους που διαχειρίζονται τα συστατικά στοιχεία της διεπαφής. Η κλάση αυτή ονομάζεται `controller` και μπορεί να οριστεί μέσω του `Scene Builder`, το αποτέλεσμα του οποίου, καταχωρείται στο αρχείο `FXML`:

```
<BorderPane fx:controller="project.MainController"
```

Για την ακρίβεια, η κλάση αυτή ορίζεται στο ανώτατο ιεραρχικά στοιχείο που υπάρχει στη διεπαφή. Στην προκειμένη περίπτωση είναι ένα `BorderPane`.

Πλέον, ο προγραμματιστής, οποιοδήποτε στοιχείο της διεπαφής θελήσει να διαχειριστεί προγραμματιστικά, μπορεί να το κάνει, δημιουργώντας στην κλάση `controller` ένα αντικείμενο με τον τύπο του στοιχείου και με όνομα, ίδιο με αυτό της ετικέτας `id`, που έχει δώσει νωρίτερα στο στοιχείο, στο αρχείο `FXML`.

4.2.2 Διαχείριση (events) γεγονότων

Προκειμένου ο προγραμματιστής να διαχειριστεί τα γεγονότα με τα οποία ο χρήστης αλληλεπιδρά με την εφαρμογή, η `JavaFX` παρέχει δύο τρόπους, ανάλογα την περίπτωση και την τακτική που ακολουθεί ο προγραμματιστής. Παρακάτω παρουσιάζονται οι τρόποι.

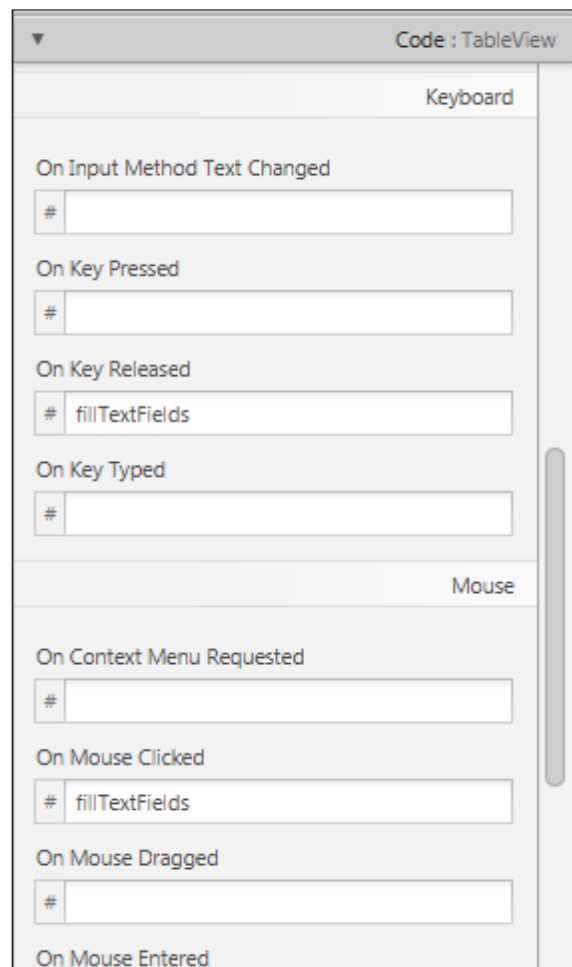
A. Μέσω του Scene Builder και την FXML

Ο προγραμματιστής αυτό που έχει να κάνει, είναι απλώς, να συμπληρώσει το όνομα της μεθόδου στο επιθυμητό event. Μόλις το event ενεργοποιηθεί, θα εκτελεστεί ο κώδικας της μεθόδου.

Ο τρόπος αυτός μπορεί να εφαρμοστεί όταν η μέθοδος που επιλέγεται δεν περιέχει παραμέτρους, καθώς σε αντίθετη περίπτωση, δε γίνεται να δηλωθεί η μέθοδος στην FXML.

Με τον τρόπο αυτό υπάρχει η δυνατότητα να μείνει πιο “καθαρός” ο κώδικας της κλάσης controller, καθώς περιέχει μόνο τις μεθόδους και όχι τον κώδικα που ορίζει σε ποιο σημείο αυτές θα εκτελούνται.

Στο παράδειγμα που παρουσιάζεται παρακάτω, ορίζεται πως η μέθοδος με όνομα “fillTextFields” θα εκτελείται κάθε φορά που στο συγκεκριμένο tableView θα απελευθερώνεται ένα πλήκτρο από το πληκτρολόγιο.



```
<TableView fx:id="table" onKeyReleased="#fillTextFields"
```

B. Προγραμματιστικά

Ο δεύτερος τρόπος διαχείρισης των γεγονότων είναι καθαρά προγραμματιστικός. Σε αυτήν την περίπτωση, το στοιχείο της διεπαφής με το οποίο αλληλεπιδρά ο χρήστης, θα πρέπει να φέρει ένα αναγνωριστικό σύμφωνα με το οποίο ο προγραμματιστής θα μπορέσει να του ορίσει τον κατάλληλο χειριστή γεγονόςτος. Ο τρόπος για να οριστεί το id στον Scene Builder είναι αντίστοιχος με αυτόν της μεθόδου για ένα γεγονός. Από τη μεριά του κώδικα, θα πρέπει να δημιουργηθεί ένα αντικείμενο με όνομα, αυτό του id και τύπο, τον τύπο του στοιχείου.

Πριν από κάθε αντικείμενο που αναπαριστά ένα στοιχείο της διεπαφής, θα πρέπει να υπάρχει το annotation @FXML.

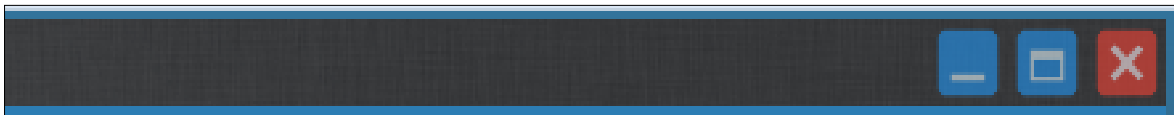
Στην περίπτωση αυτή, που η διαχείριση των γεγονότων γίνεται προγραμματιστικά, μία εναλλακτική λύση είναι τα expression Lambda τα οποία περιορίζουν τις γραμμές του κώδικα και τον κάνουν πιο ξεκάθαρο χωρίς καμία αλλαγή στο αποτέλεσμα.

```
@FXML private ImageView imageView;

imageView.setOnMouseClicked( (MouseEvent) -> {
    insertImage();
});
imageView.setOnMouseEntered( (MouseEvent) -> {
    plusOpacity();
});
imageView.setOnMouseExited( (MouseEvent) -> {
    minusOpacity();
});
```

4.2.3 Δημιουργία νέων Windows Borders

Στο πλαίσιο της πτυχιακής εργασίας, αλλά και για λόγους αισθητικούς, θέλησα να προσθέσω νέα borders στη διεπαφή της εφαρμογής μέσω της JavaFX. Παρακάτω γίνεται μία σύντομη περιγραφή του τρόπου με το οποίο δημιουργήθηκαν. Ο τρόπος αυτός έχει εφαρμοστεί σε όλα τα παράθυρα της εφαρμογής.



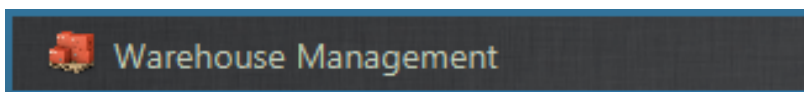
Ωστόσο, οι ιδιότητες των νέων borders δεν είναι ολοκληρωμένες, όπως αυτές παρέχονται από αυτά των άλλων λειτουργικών, καθώς η μόνη ιδιότητα που φέρουν είναι η μετακίνηση των παραθύρων μέσω της ιδιότητας drag & drop. Η JavaFX ωστόσο δίνει τη δυνατότητα να αποκτήσουν πλήρη λειτουργικότητα. Από κει και πέρα, τα νέα borders φέρουν τα τρία γνωστά πλήκτρα ελαχιστοποίησης, μεγιστοποίησης και κλεισίματος του παραθύρου.

Για τη δημιουργία των borders στο αρχικό παράθυρο της εφαρμογής, έχουν δημιουργηθεί σε σειρά ιεραρχίας:

- **1 AnchorPane:** Βρίσκεται στην κορυφή του παραθύρου και περιέχει όλα τα παρακάτω. Ο λόγος για τον οποίο έγινε η επιλογή του AnchorPane είναι η ιδιότητα του να “αγκυροβολεί” τα αντικείμενα που περιέχει σε συγκεκριμένα σημεία, γεγονός το οποίο εξυπηρετεί στη στοίχιση των αντικειμένων.

```
AnchorPane.setRightAnchor(groupButton, 0.0);
```

- **1 Label:** Περιέχει το όνομα της εφαρμογής.



- **1 Group:** Περιέχει τα τρία κουμπιά διαχείρισης του παραθύρου.



- **4 ImageView:** Το πρώτο περιέχει το icon της εφαρμογής όπως φαίνεται παραπάνω. Τα υπόλοιπα τρία λειτουργούν ως Buttons και αποτελούν τα τρία κουμπιά.

4.2.3.1 Drag & Drop παραθύρου

Βασική ιδιότητα των borders είναι η δυνατότητα μετακίνησης του παραθύρου μέσω της λειτουργίας drag & drop. Αυτό επιτυγχάνεται μέσω της JavaFX με τον εξής τρόπο.

```
header.setOnMousePressed((MouseEvent mouseEvent) -> {  
    pos_x = Main.mainStage.getX() - mouseEvent.getScreenX();  
    pos_y = Main.mainStage.getY() - mouseEvent.getScreenY();  
});  
  
header.setOnMouseDragged((MouseEvent mouseEvent) -> {  
    Main.mainStage.setX(mouseEvent.getScreenX() + pos_x);  
    Main.mainStage.setY(mouseEvent.getScreenY() + pos_y);  
});
```

Η μεταβλητή header αντιπροσωπεύει το AnchorPane και η mainStage το παράθυρο. Όταν γίνεται το κλικ πάνω στο AnchorPane (event: setOnMousePressed), αποθηκεύονται στις μεταβλητές pos_x και pos_y η διαφορά των συντεταγμένων του παραθύρου και του κέρσορα. Όσο παραμένει πατημένος ο κέρσορας και μετακινείται ταυτόχρονα (event: setOnMouseDragged), οι

συντεταγμένες του παραθύρου συνεχώς ενημερώνονται με βάση τις τρέχουσες συντεταγμένες του κέρσορα και των τιμών των pos_x και pos_y.

4.2.4 Διαχείριση φωτογραφιών προϊόντων

Εκτός των άλλων στοιχείων, κάθε προϊόν διαθέτει μία φωτογραφία η οποία αποθηκεύεται στη βάση δεδομένων στη μορφή "bytea". Η εφαρμογή αναλαμβάνει να απεικονίσει την φωτογραφία του κάθε προϊόντος αλλά και να αποθηκεύσει μία φωτογραφία την οποία επέλεξε ο χρήστης για ένα νέο προϊόν.

4.2.4.1 Αποθήκευση φωτογραφίας στη βάση δεδομένων

Αρχικά, εμφανίζεται ένα παράθυρο (FileChooser) μέσω του οποίου ο χρήστης επιλέγει τη φωτογραφία που επιθυμεί. Μόλις επιλεγεί, η φωτογραφία αποθηκεύεται αρχικά σε μία μεταβλητή τύπου "File". Αφού ελεγχθεί ότι το αρχείο είναι έγκυρο, μέσω της διεύθυνσης του περνά σε μία μεταβλητή τύπου "FileInputStream" μέσα από την οποία μπορούμε να διαβάσουμε τα δεδομένα του αρχείου της φωτογραφίας. Από κει και πέρα, το κατάλληλο prepared statement αναλαμβάνει να κάνει την εγγραφή στη βάση μέσω της μεθόδου "setBinaryStream".

```
prestatement.setBinaryStream(6, (InputStream)fis, (int)selectedFile.length());
```

4.2.4.2 Ανάκτηση και απεικόνιση φωτογραφίας

Με αντίστοιχο τρόπο γίνεται και η ανάκτηση της φωτογραφίας από μία βάση δεδομένων. Κάνοντας το κατάλληλο sql ερώτημα στη βάση, μαζί με όλα τα άλλα, ανακτώνται και οι φωτογραφίες των προϊόντων. Μέσω της μεθόδου "getBinaryStream" του ResultSet αντικειμένου, τα δεδομένα της φωτογραφίας αποθηκεύονται σε μία μεταβλητή τύπου InputStream. Από κει και πέρα ο προγραμματιστής μπορεί να διαχειριστεί τη φωτογραφία, όπως αυτός θέλει.

```
InputStream is = rs.getBinaryStream("φωτογραφία");  
  
Image image = new Image(is);  
is.close();  
  
imageView.setImage(image);
```

4.2.5 Περιγραφή μεθόδων εφαρμογής

Στο σημείο αυτό γίνεται περιγραφή μερικών δυνατοτήτων της εφαρμογής που ως σκοπό έχει την ανάδειξη κάποιων χαρακτηριστικών της JavaFX.

4.2.5.1 Αναζήτηση προϊόντος

Στο κύριο παράθυρο της εφαρμογής παρέχεται στο χρήστη η δυνατότητα αναζήτησης κάποιου-ων προϊόντων της βάσης, τα οποία εμπεριέχουν σε κάποιο από τα πεδία τους (Κωδικός, Περιγραφή, Επωνυμία) τη φράση που υπάρχει στο πεδίο αναζήτησης.

	Κωδικός	Περιγραφή
1L	1001004	AMITA MOTION 1L
	1001006	AMITA ANANAS 1L

Αυτό επιτυγχάνεται μέσω του γεγονότος “OnKeyReleased”. Κάθε φορά που ο χρήστης πληκτρολογεί κάτι στο πεδίο, ο κεντρικός πίνακας ανανεώνεται με τα νέα αποτελέσματα.

4.2.5.2 Ανανέωση φόρμας προϊόντος

Με αντίστοιχο τρόπο ενημερώνονται και τα πεδία στα αριστερά του παραθύρου που φέρουν τα χαρακτηριστικά κάποιου προϊόντος. Κάθε φορά που επιλέγεται ένα προϊόν από τον πίνακα, τα πεδία λαμβάνουν τα χαρακτηριστικά του προϊόντος.

Κωδικός	1001033	COCA COLA ZERO 1L
1001090	1001056	ARIZONA ICE BLACK TEA 0.591L
	1001089	VIVA CRANBERRY 1L
Περιγραφή	1001090	VIVA FRESH ΜΗΛΟ 1L
VIVA FRESH ΜΗΛΟ 1L	1001091	VIVA ΧΥΜΟΣ ΠΟΡΤΟΚΑΛΙ ΝΕΚΤΑΡ 1L
Επωνυμία	1001092	VIVA ΧΥΜΟΣ ΠΟΡΤΟΚΑΛΙ ΜΗΛΟ ΡΟΔΑΚΙΝΟ 1L
VIVA	1001093	VIVA ΧΥΜΟΣ ΒΥΣΣΙΝΟ 1L
	1001094	VIVA ΧΥΜΟΣ ΕΧΟΤΙC 1L
Είδος	1001095	VIVA ΧΥΜΟΣ ΑΝΑΝΑΣ 1L
ΑΝΑΨΥΚΤΙΚΑ-RDT'S	1001096	VIVA ΧΥΜΟΣ ΛΕΜΟΝΙ 1L

4.2.5.3 Εισαγωγή ακροατών στην εισαγωγή δεδομένων

Όπως προαναφέρθηκε, έχει δοθεί μεγάλη προσοχή στην εφαρμογή, όσο αφορά την εισαγωγή των δεδομένων από το χρήστη. Γενικώς στην εφαρμογή, στα περισσότερα σημεία όπου ο χρήστης μπορεί να εισάγει δεδομένα, η εφαρμογή καθοδηγεί το χρήστη μέσω των ακροατών (listeners).

Ένας ακροατής μπορεί μεταξύ άλλων να ορίσει συγκεκριμένα πρότυπα δεδομένων τα οποία είναι επιτρεπτά. Ορίζονται συνήθως στην μέθοδο initialize η οποία εκτελείται με την εκκίνηση της εφαρμογής. Ουσιαστικά, από τη στιγμή που ορίζεται, “ακούει” αυτά που γίνονται στο στοιχείο το οποίο έχει “αναλάβει”.

```
searchField.textProperty().addListener((ov, oldValue, newValue) -> {
    searchField.setText(newValue.toUpperCase());
});
```

Ο συγκεκριμένος ακροατής μετατρέπει οποιονδήποτε χαρακτήρα σε κεφαλαίο, σε περίπτωση που αυτό είναι πεζός.

```
apothemaField.textProperty().addListener((ov, oldValue, newValue) -> {
    if(!newValue.matches("[0-9]{0,4}"))
        apothemaField.setText(oldValue);
});
```

Ο ακροατής αυτός επιτρέπει την εισαγωγή μόνο αριθμητικών χαρακτήρων με πλήθος ψηφίων 0-4.

Όσο αφορά τη επιθυμητή εισαγωγή δεδομένων, ένας καλός τρόπος είναι η μέθοδος “matches” που προσφέρει η Java, στην οποία ο προγραμματιστής περνά ως παράμετρο μία κανονική έκφραση βάση της οποίας, φιλτράρονται τα δεδομένα.

```
if(eidosTextField.getText().matches("[A-Ω0-9A-Z' *,.&'%]{1,30}")) {
```

Το παραπάνω if-clause εκτελείται μόνο στην περίπτωση όπου το περιεχόμενο του “eidosTextField” περιέχει ελληνικούς ή αγγλικούς κεφαλαίους χαρακτήρες, αριθμητικούς χαρακτήρες ή και κάποιους από τους χαρακτήρες *,.&%', συνολικού μήκους 0-30.

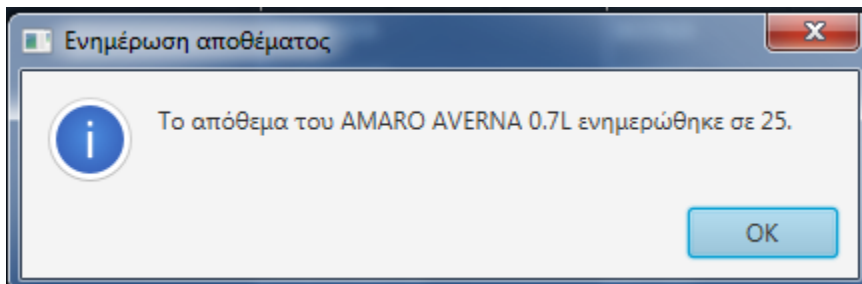
4.2.5.4 JavaFX Dialogs

Η JavaFX από την έκδοση 8 υποστηρίζει τα δικά της Dialogs. Μέσω της κλάσης Alert ο χρήστης μπορεί να επικοινωνήσει με την εφαρμογή. Είτε μέσω ενός ενημερωτικού μηνύματος της εφαρμογής, είτε μέσω μίας επιλογής ή επιβεβαίωσης που ζητά η εφαρμογή από το χρήστη.

Η εισαγωγή τους στον κώδικα της εφαρμογής είναι πολύ απλή και είναι άκρως παραμετροποιήσιμα. Εκτός της κλάσης Alert υπάρχουν και άλλες κλάσεις όπως η TextInputDialog, η ChoiceDialog κ.α.

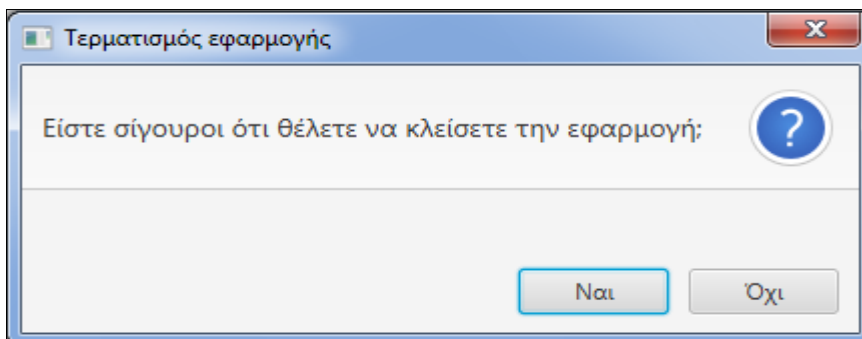
Στην παρούσα εφαρμογή έχουν χρησιμοποιηθεί τρεις τύποι της κλάσης Alert.

INFORMATION: Χρησιμοποιούνται συνήθως για να κάνουν μία απλή ενημέρωση στο χρήστη.



```
Alert al = new Alert(Alert.AlertType.INFORMATION);
al.setTitle("Ενημέρωση αποθέματος");
al.setHeaderText("");
al.setContentText("Το απόθεμα του " + perigrafhProiontos +
                  " ενημερώθηκε σε " + apothemaProiontos + ".");
al.showAndWait();
```

CONFIRMATION: Χρησιμοποιείται για να λάβει η εφαρμογή μία επιβεβαίωση από το χρήστη. Ωστόσο, υπάρχει η δυνατότητα να παραμετροποιηθεί, ούτως ώστε η επιλογή να είναι παραπάνω από δύο.



```
Alert all = new Alert(Alert.AlertType.CONFIRMATION);
all.setTitle("Τερματισμός εφαρμογής");
all.setHeaderText("Οι ημερήσιες πωλήσεις είναι ακόμα ανοιχτές");
all.setContentText("Είστε σίγουροι ότι θέλετε να κλείσετε την εφαρμογή;");

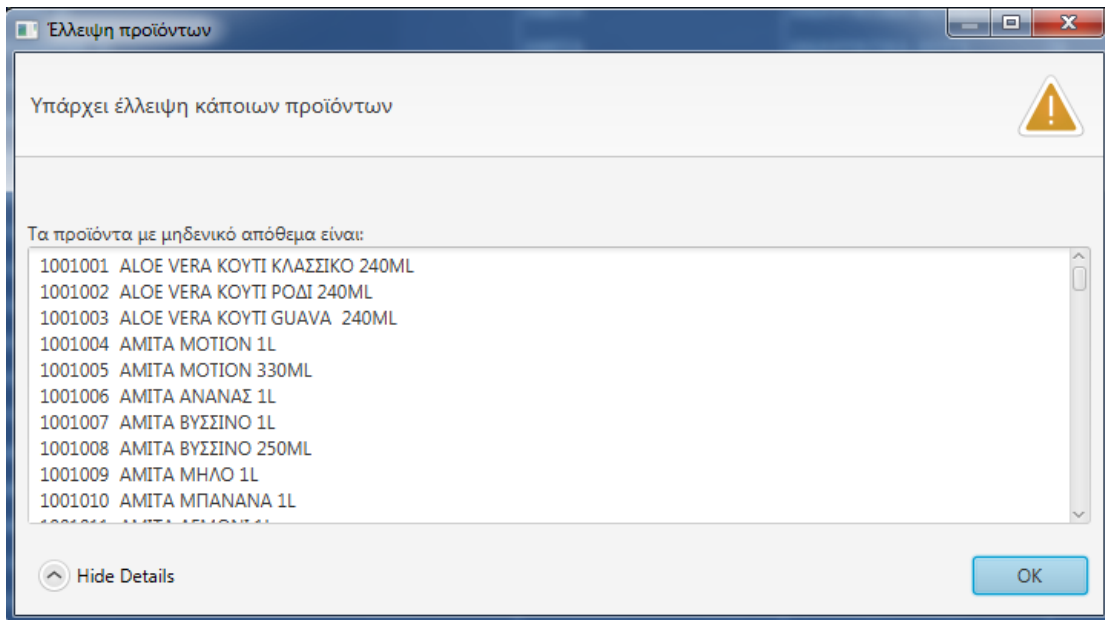
ButtonType buttonTypeYes = new ButtonType("Ναι");
ButtonType buttonTypeNo = new ButtonType("Όχι");

all.getButtonTypes().setAll(buttonTypeYes, buttonTypeNo);

Optional<ButtonType> result = all.showAndWait();

if(result.get() == buttonTypeYes) {
    Platform.exit();
}
else {
    MainController.dailySalesStage.toFront();
}
```


WARNING: Χρησιμοποιείται για να προειδοποιήσει η εφαρμογή τον χρήστη για κάποιο γεγονός. Η ιδιότητα του μπορεί να παρομοιαστεί με αυτή του INFORMATION. Στην παρούσα εφαρμογή, σε ένα από τα σημεία που χρησιμοποιείται, ενημερώνει το χρήστη για τα προϊόντα τα οποία δεν έχουν απόθεμα στην αποθήκη. Αυτό γίνεται με το που ξεκινάει η εφαρμογή.



Ουσιαστικά, μέσα στο κύριο σώμα του παραθύρου, δημιουργείται ένα textArea το οποίο μέσω ενός sql ερωτήματος στη βάση, επιστρέφει τον κωδικό και την περιγραφή των προϊόντων με μηδενικό απόθεμα.

```
Alert alert = new Alert(Alert.AlertType.WARNING);
alert.setTitle("Έλλειψη προϊόντων");
alert.setHeaderText("Υπάρχει έλλειψη κάποιων προϊόντων");
alert.setContentText("");

Label label = new Label("Τα προϊόντα με μηδενικό απόθεμα είναι:");

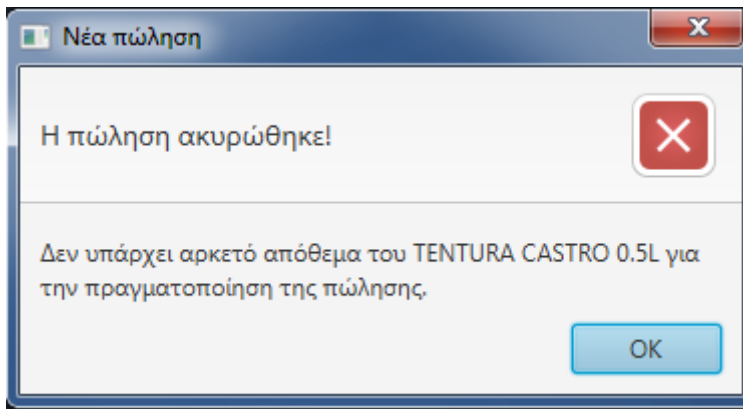
TextArea textArea = new TextArea(perigrafes);
textArea.setEditable(false);
textArea.setWrapText(true);

textArea.setMaxWidth(Double.MAX_VALUE);
textArea.setMaxHeight(Double.MAX_VALUE);
GridPane.setVgrow(textArea, Priority.ALWAYS);
GridPane.setHgrow(textArea, Priority.ALWAYS);

GridPane expContent = new GridPane();
expContent.setMaxWidth(Double.MAX_VALUE);
expContent.add(label, 0, 0);
expContent.add(textArea, 0, 1);

alert.getDialogPane().setExpandableContent(expContent);
alert.showAndWait();
```

ERROR: Και αυτός ο τύπος μπορεί να παρομοιαστεί με τους προηγούμενους δύο. Ρόλος αυτού είναι να ενημερώσει το χρήστη για κάποιο σφάλμα.



Στην προκειμένη περίπτωση, το σύστημα δεν επιτρέπει το χρήστη να πραγματοποιήσει την πώληση, καθώς δεν υπάρχει επαρκές απόθεμα για να πραγματοποιηθεί αυτή.

```
Alert al = new Alert(Alert.AlertType.ERROR);
al.setTitle("Νέα πώληση");
al.setHeaderText("Η πώληση ακυρώθηκε!");
al.setContentText("Δεν υπάρχει αρκετό απόθεμα του " + perigraphField.getText() +
    " για την πραγματοποίηση της πώλησης.");
al.showAndWait();
```

4.2.5.5 Προγραμματιστική μορφοποίηση του στυλ

Ένα άλλο χαρακτηριστικό της JavaFX είναι το γεγονός ότι επιτρέπει τον προγραμματιστή να μορφοποιεί το στυλ της διεπαφής μέσα από το κώδικα. Αυτό βοηθά στο γεγονός ότι μεγαλώνει η αίσθηση αλληλεπίδρασης της εφαρμογής με το χρήστη. Στην εφαρμογή, αυτό γίνεται πράξη στο σημείο όπου ο χρήστης εισάγει τα στοιχεία ενός νέου προϊόντος. Τη στιγμή όπου ο χρήστης πληκτρολογεί τα στοιχεία, μόλις εισαχθεί ένας μη έγκυρος χαρακτήρας, το περίγραμμα του πεδίου γίνεται κόκκινο. Αντίθετα, όσο όλοι οι χαρακτήρες είναι έγκυροι παραμένει πράσινο.



Μέσω της μεθόδου “setStyle” η οποία υλοποιείται σε όλα τα στοιχεία της κλάσης Node, ο προγραμματιστής μπορεί να εισάγει ένα νέο κανόνα CSS περνώντας τον ως παράμετρο στη μέθοδο.

```
if(perigraphhTextField.getText().matches("[Α-Ω0-9Α-Ζ' *,.&-'%]{1,50}")) {  
    flag3 = true;  
    perigraphhMessage.setText("");  
    perigraphhTextField.setStyle("-jfx-focus-color:green");  
}  
else {  
    flag3 = false;  
    perigraphhMessage.setText("~Επιτρεπόμενοι χαρακτήρες: *,.&-'%");  
    perigraphhTextField.setStyle("-jfx-focus-color:red");  
}
```

4.3 Εναλλακτικοί τρόποι υλοποίησης της εφαρμογής

Ο εναλλακτικός τρόπος υλοποίησης έχει να κάνει τόσο με τη μεριά του κώδικα, δηλαδή τον προγραμματιστικό τρόπο υλοποίησης, όσο και με το τελικό αποτέλεσμα το οποίο εμφανίζεται στο χρήστη και με το οποίο τελικά αλληλεπιδρά.

4.3.1 Ολοκλήρωση πώλησης στην περίπτωση μηδενικού αποθέματος

Η εφαρμογή μέσω της αντίστοιχης σελίδας, επιτρέπει στο χρήστη να διαχειρίζεται τις ημερήσιες πωλήσεις του, ενημερώνοντας άμεσα το απόθεμα στην αποθήκη μετά την πραγματοποίηση μίας πώλησης.

Σε περίπτωση όπου το απόθεμα ενός προϊόντος είναι μηδενικό, όταν ο χρήστης πάει να πραγματοποιήσει μία πώληση αυτού του προϊόντος, η εφαρμογή δε επιτρέπει την ολοκλήρωση της, ενημερώνοντας το χρήστη πως δεν υπάρχει απόθεμα αυτού του προϊόντος.

Για λόγους πρακτικούς όμως και μόνο, το σύστημα θα μπορούσε να επιτρέψει στο χρήστη να ολοκληρώσει την πώληση, δηλαδή να καταχωρηθεί στις ημερήσιες πωλήσεις χωρίς φυσικά να ενημερωθεί το απόθεμα στη βάση. Αυτό θα είχε νόημα, στην περίπτωση όπου, στο κατάστημα έχει γίνει παραλαβή νέας προμήθειας και ο χρήστης του συστήματος δεν έχει προλάβει να ενημερώσει τη βάση με τα νέα αποθέματα των προϊόντων.

4.3.2 Δυνατότητα επεξεργασίας των στοιχείων κάθε προϊόντος από το αριστερό μενού της κύριας σελίδας

Στην κύρια σελίδα της εφαρμογής γίνεται προεπισκόπηση των προϊόντων της αποθήκης. Στο αριστερό μέρος υπάρχουν κάποια πεδία τα οποία φέρουν τα στοιχεία ενός προϊόντος. Το προϊόν που προβάλλεται, επιλέγεται από το χρήστη κάθε φορά που αλλάζει το επιλεγμένο στον πίνακα προϊόν.

Η εφαρμογή στην παρούσα φάση δεν επιτρέπει την επεξεργασία των στοιχείων του προϊόντος (εκτός του αποθέματος) μέσω των πεδίων αυτών. Γι' αυτό, σε περίπτωση που ο χρήστης θελήσει να αλλάξει κάποιο από αυτά, θα πρέπει να διαγράψει και να καταχωρήσει εκ νέου το προϊόν με τα ενημερωμένα στοιχεία.

Αυτό μπορεί να παρακαμφθεί αν δίνεται η δυνατότητα επεξεργασίας τους, όπως συμβαίνει ήδη στο πεδίο του αποθέματος. Για να αποφευχθούν μη έγκυρες εισαγωγές δεδομένων, μπορούν να χρησιμοποιηθούν τα validations που χρησιμοποιούνται και στην δημιουργία νέου προϊόντος.

4.3.3 Δημιουργία προσαρμοσμένου Tooltip

Στη δευτερεύουσα διεπαφή της εφαρμογής όπου παρουσιάζονται πληροφορίες σχετικά με την εφαρμογή, χρησιμοποιείται το στοιχείο Tooltip μέσω του οποίου η εφαρμογή ενημερώνει το χρήστη μέσω ενός αναδυόμενου πεδίου για κάποιο στοιχείο στο οποίο ο χρήστης έχει αφήσει το δείκτη πάνω.

Το κακό με το στοιχείο αυτό είναι ότι η καθυστέρηση μέχρι να εμφανιστεί είναι αρκετά μεγάλη. Η JavaFX δεν ορίζει κάποια μέθοδο μέσω της οποίας να μπορείς να ορίσεις το δικό σου χρόνο καθυστέρησης μέχρι την εμφάνιση του Tooltip. Ωστόσο, έχουν γίνει κάποιες προσπάθειες από κάποια άτομα για τη δημιουργία κάποιων μεθόδων μέσω των οποίων μπορείς να χειριστείς και να ορίσεις το δικό σου χρόνο καθυστέρησης.

4.4 Προτάσεις ανάπτυξης της εφαρμογής

Αρχικά είχε αναφερθεί πως για τους σκοπούς της πτυχιακής εργασίας, η εφαρμογή παρέχει κάποιες δυνατότητες οι οποίες επικεντρώνονται στη διαχείριση των προϊόντων μίας αποθήκης και μόνο. Επίσης, κάποιες άλλες δυνατότητες έχουν αποδοθεί σε συγκεκριμένα σημεία και μόνο, και όχι σε όλα τα σημεία στα οποία η εφαρμογή τους είναι λειτουργική. Ωστόσο, εφαρμογές αυτού του είδους

και οι οποίες προορίζονται για επαγγελματική και εμπορική χρήση, είναι απαραίτητο να πληρούν όλες τις βασικές ιδιότητες οι οποίες έχουν να κάνουν με τη διαχείριση μίας αποθήκης για να θεωρηθούν ολοκληρωμένες.

4.4.1 Διαχείριση προμηθευτών

Μία βασική λειτουργία που μπορεί να φέρει μία εφαρμογή σαν και αυτή είναι η διαχείριση των προμηθευτών και των παραγγελιών που γίνονται προς αυτούς. Στη βάση θα πρέπει να υπάρχουν πίνακες για τους προμηθευτές και τις παραγγελίες μέσω των τιμολογίων του καταστήματος. Έτσι, ο χρήστης θα μπορεί να διαχειρίζεται καλύτερα τα στοιχεία των παραγγελιών και γενικώς τις συναλλαγές με τους προμηθευτές.

4.4.2 Αυτόματη δημιουργία παραγγελίας

Η εφαρμογή θα μπορούσε να έχει τη δυνατότητα να δημιουργεί αυτόματα μία νέα παραγγελία προς κάποιον προμηθευτή, ελέγχοντας τα αποθέματα των προϊόντων, ορίζοντας πρώτα το ελάχιστο επιθυμητό απόθεμα για κάθε προϊόν ξεχωριστά.

Αυτό θα εξυπηρετούσε αρκετά το χρήστη, ειδικά σε καταστήματα όπου υπάρχει αρκετά μεγάλη αγοραπωλησία καθημερινά, καθώς τον γλιτώνει από το χρόνο αναζήτησης των προϊόντων προς παραγγελία και γενικώς της δημιουργίας μίας παραγγελίας, αποφεύγοντας ταυτόχρονα τυχόν λάθη στον αριθμό των κομματιών και στην περιγραφή του προϊόντος.

4.4.3 Καταχώρηση πελατών

Σε περίπτωση όπου το κατάστημα θέλει για παράδειγμα να επιβραβεύσει τους πελάτες του για κάποιες ενέργειες όπως, συμπλήρωση ενός αριθμού αγορών από το κατάστημα, αγορά προϊόντων τα οποία κάνει το κατάστημα ειδική προώθηση κ.α. θα ήταν απαραίτητη η δυνατότητα του συστήματος να καταχωρεί στη βάση δεδομένων του τα βασικά στοιχεία των πελατών.

4.5 Σύνοψη κεφαλαίου

Το τέταρτο και τελευταίο μέρος της πτυχιακής εργασίας, παρείχε τη δυνατότητα στο αναγνώστη, να πάρει μία πρώτη εικόνα από το σύνολο της εφαρμογής.

Αρχικά έγινε η παρουσίαση της διεπαφής, ενώ στη συνέχεια έγινε αναφορά στα σημαντικότερα σημεία του κώδικα της εφαρμογής. Σημαντικά, τόσο για τη λειτουργία μίας JavaFX εφαρμογής, όσο και για την ανάδειξη κάποιων σημαντικών ιδιοτήτων της JavaFX.

Τέλος, παρουσιάστηκαν κάποιοι εναλλακτικοί τρόποι με τους οποίους θα μπορούσε η εφαρμογή να υλοποιηθεί καθώς και προτάσεις περαιτέρω ανάπτυξης και ολοκλήρωσης της εφαρμογής.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η JavaFX πρόκειται να αποτελέσει το βασικότερο εργαλείο για έναν ο οποίος πρόκειται ασχοληθεί με τη δημιουργία διεπαφών πάνω στην Java. Οι δυνατότητες που παρέχει, επιτρέπουν τον προγραμματιστή να πραγματοποιήσει κάθε του ιδέα πάνω σε μία εφαρμογή. Με την παράλληλη υποστήριξη που παρέχει η Oracle στη JavaFX μέσω των επανεκδόσεων της και της συνεχής παρακολούθησης των τεχνολογικών τάσεων, ο προγραμματιστής, μπορεί να παρέχει στον πελάτη ένα πλήρως λειτουργικό προϊόν, φιλικό προς το χρήστη, ικανοποιώντας κάθε του ανάγκη.

Αλλά και για αυτόν που ασχολείται ήδη με το αντικείμενο αυτό και ο οποίος έχει εξοικειωθεί με άλλες πλατφόρμες λογισμικού όπως η Swing, η Java παρέχει τη δυνατότητα σε αυτόν να συνεχίσει να τη χρησιμοποιεί, επιτρέποντας του να ενσωματώσει υλικό αυτής μέσα σε μία εφαρμογή βασισμένη στη JavaFX και το αντίθετο. Με τον τρόπο αυτό ο προγραμματιστής μπορεί να μεταβεί στη νέα τεχνολογία με πολύ πιο ομαλό τρόπο.

ΑΝΑΦΟΡΕΣ

- [1] JavaFX Introduction,
https://www.ntu.edu.sg/home/ehchua/programming/java/Javafx1_intro.html,
(τελευταία πρόσβαση 29.11.2017).
- [2] Fedortsova I. (2013), Embedding Swing Content in JavaFX Applications,
https://docs.oracle.com/javafx/8/embed_swing/jfxpub-embed_swing.htm,
(τελευταία πρόσβαση 13.12.2017).
- [3] Retina Display,
https://en.wikipedia.org/wiki/Retina_Display,
(τελευταία πρόσβαση 29.12.2017).
- [4] iText,
<https://en.wikipedia.org/wiki/IText>,
(τελευταία πρόσβαση 10.1.2018).
- [5] Material Design,
https://el.wikipedia.org/wiki/Material_design,
(τελευταία πρόσβαση 11.1.2018).
- [6] Ahmad, S. (2017), JavaFX Material Design Setup Tutorial with JFoenix library,
<https://shahsparx.me/javafx-material-design-jfoenix/>,
(τελευταία πρόσβαση 11.1.2018).

ΒΙΒΛΙΟΓΡΑΦΙΑ

Αντάμης, Ε. (2015), Ανάπτυξη λογισμικού android: Σύστημα απομακρυσμένης παραγγελιοληψίας με χρήση συσκευών κινητής τηλεφωνίας, Βόλος.

Σπέντζας, Κ. (2017), Χρήση του JavaFX για τη δημιουργία μιας εφαρμογής Java δομημένης σε επίπεδα, Λάρισα.

Anderson, G. & Anderson, P. (2014), JavaFX Rich Client Programming on the NetBeans Platform.

Sharan, K. (2015), Learn JavaFX 8: Building User Experience and Interfaces with Java 8.

Taman, M. (2015), JavaFX Essentials.

Lowe, D., 10 Differences between JavaFX and Swing,
<http://www.dummies.com/programming/java/10-differences-between-javafx-and-swing/> (τελευταία πρόσβαση 28.12.2017).

Pawlan, M. (2013), What is JavaFX?,
<https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>,
(τελευταία πρόσβαση 17.11.2017).

Getting Started with JavaFX: JavaFX Overview,
<https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>,
(τελευταία πρόσβαση 23.11.2017).

Getting Started with JavaFX: Understanding the JavaFX Architecture,
<https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-architecture.htm>,
(τελευταία πρόσβαση 19.11.2017).

The JavaFX Advantage for Swing Developers,
<https://docs.oracle.com/javase/8/javafx/interoperability-tutorial/overview.htm>,
(τελευταία πρόσβαση 12.10.2017).